Numerical methods for conservation laws and related equations

Siddhartha Mishra, Ulrik Skre Fjordholm and Rémi Abgrall

About these notes

These notes present numerical methods for conservation laws and related time-dependent nonlinear partial differential equations. The focus is on both simple scalar problems as well as multi-dimensional systems.

The MATLAB package Compack (COnservation law Matlab PACKage) has been developed as an educational tool to be used with these notes. All the numerical experiments in the lecture notes have been performed in Compack. The scripts used to generate figures and tables are all in the `+Notes` subpackage. For instance, to generate the plots in Figure 2.3, run `Notes.Chapter2.central()` from the Compack base folder. Figures are saved to the `output` folder. Compack can be downloaded from

`https://github.com/ulriksf/compack`

Contents

CHAPTER 1

Introduction

Many interesting problems in the physical, biological, engineering and social sciences are modeled by a simple paradigm: Consider a domain $\Omega \subset \mathbb{R}^n$ and a quantity of interest $\mathbf{U}$, defined for all points $\mathbf{x} \in \Omega$. The quantity of interest $\mathbf{U}$ may be the temperature of a rod, the pressure of a fluid, the concentration of a chemical or a group of cells or the density of a human population. The evolution (in time) of this quantity of interest $\mathbf{U}$ can be described by a simple phenomenological observation:

> *The temporal rate of change of* $\mathbf{U}$ *in any fixed sub-domain* $\omega \subset \Omega$ *is equal to the total amount of* $\mathbf{U}$ *produced or destroyed inside* ω *and the flux of* $\mathbf{U}$ *across the boundary* $\partial\omega$.

The above observation says that the change in $\mathbf{U}$ is due to two factors: the *source* or *sink*, representing the quantity produced or destroyed, and the *flux*, representing the amount of $\mathbf{U}$ that either goes in or comes out of the sub-domain, see Figure 1.1. This observation is mathematically rendered as

$$\frac{d}{dt}\int_\omega \mathbf{U}\, d\mathbf{x} = \underbrace{-\int_{\partial\omega} \mathbf{F}\cdot\nu\, d\sigma(\mathbf{x})}_{\text{flux}} + \underbrace{\int_\omega \mathbf{S}\, d\mathbf{x}}_{\text{source}}, \tag{1.1}$$

where ν is the unit outward normal, $d\sigma(\mathbf{x})$ is the surface measure, and $\mathbf{F}$ and $\mathbf{S}$ are the flux and the source respectively. The minus sign in front of the flux term is for convenience. Note that (1.1) is an *integral* equation for the evolution of the total amount of $\mathbf{U}$ in ω.

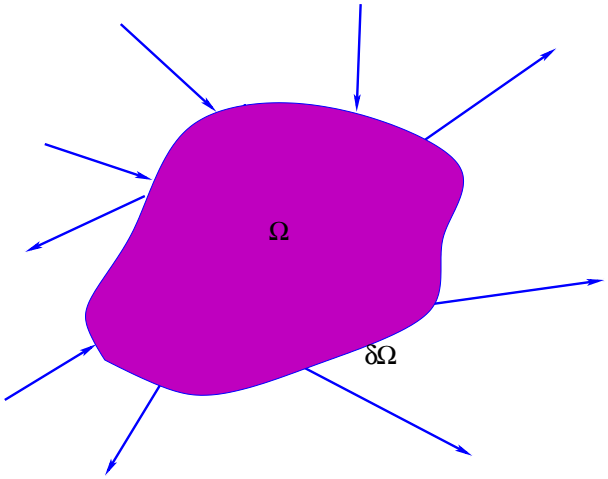


FIGURE 1.1. An illustration of conservation in a domain with the change being determined by the net flux.

We simplify (1.1) by using integration by parts (or the Gauss divergence theorem) on the surface integral to obtain

$$\frac{d}{dt}\int_\omega \mathbf{U}\, d\mathbf{x} + \int_\omega \operatorname{div}(\mathbf{F})\, d\mathbf{x} = \int_\omega \mathbf{S}\, d\mathbf{x}. \tag{1.2}$$

Since (1.2) holds for all sub-domains ω of Ω, we can use an infinitesimal ω to obtain the following *differential* equation:

$$\mathbf{U}_t + \operatorname{div}(\mathbf{F}) = \mathbf{S} \qquad \forall\ (\mathbf{x}, t) \in (\Omega, \mathbb{R}_+). \tag{1.3}$$

The differential equation (1.3) is often termed as a *balance law* as it is a statement of the fact that the rate of change in $\mathbf{U}$ is a balance of the flux and the source. Frequently, the only change in $\mathbf{U}$ is from the fluxes and the source is set to zero. In such cases, (1.3) reduces to

$$\mathbf{U}_t + \operatorname{div}(\mathbf{F}) = 0 \qquad \forall\ (\mathbf{x}, t) \in (\Omega, \mathbb{R}_+). \tag{1.4}$$

Equation (1.4) is termed as a *conservation law*, as the only change in $\mathbf{U}$ comes from the quantity entering or leaving the domain of interest.

The discussion so far is very general. We have not yet specified the explicit forms of $\mathbf{U}, \mathbf{F}$ and $\mathbf{S}$. In fact, the conservation law (1.4) and the balance law (1.3) are generic to a very large number of models. Explicit forms of the quantity of interest, flux and source depend on the specific model being considered. The modeling of the flux $\mathbf{F}$ is the core function of a physicist, biologist, engineer or other domain scientists. We will provides several examples to illustrate conservation laws.

1.1. Examples for conservation laws.

For simplicity of the exposition, we begin with scalar examples, i.e, the quantity of interest $\mathbf{U}$ is a scalar U.

1.1.1. Scalar transport equation. Let $\mathbf{U} = U$ denote the concentration of a chemical (for example, a pollutant in a river). Assume that the river flows with a velocity field $\mathbf{a}(\mathbf{x}, t)$ and we know the velocity field at all points in the river. The pollutant will clearly be transported in the direction of the velocity and so the flux in this case is $\mathbf{F} = \mathbf{a}U$. Since there is no production or destruction of the pollutant during the flow, the source term in (1.3) is set to zero. Consequently, the conservation law (1.4) takes the form

$$U_t + \operatorname{div}\big(\mathbf{a}(\mathbf{x}, t)U\big) = 0. \tag{1.5}$$

This equation is linear. In the simple case of one space dimension and a constant velocity field $\mathbf{a}(\mathbf{x}, t) \equiv a$, (1.5) reduces to

$$U_t + aU_x = 0. \tag{1.6}$$

The scalar one-dimensional equation (1.6) is often referred to as the transport or *advection* equation.

1.1.2. The heat equation. Another illustrative example of a conservation law is provided by heat conduction. Assume that a hot material (like a metal block) is heated at one end and is left to cool afterwards, without providing any additional source of heat. It is a common observation that the heat spreads or *diffuses* out and the temperature of the material becomes uniform after some time. Let U be the temperature of the material. Diffusion of heat is governed by Fourier's or Fick's law

$$\mathbf{F}(U) = -\mathbf{k}\nabla U.$$

Here, $\mathbf{k}$ is the conductivity tensor for the medium. The minus sign is due to the fact that heat flows from hotter to cooler zones. Substituting Fourier's law into the conservation law (1.4), we obtain the *heat* equation

$$U_t - \operatorname{div}\big(\mathbf{k}\nabla U\big) = 0. \tag{1.7}$$

If the conductivity is assumed to be unity and the material is one-dimensional (like a rod), (1.7) reduces to the well-known one-dimensional heat equation

$$U_t - U_{xx} = 0. \tag{1.8}$$

The scalar transport equation (1.5) and the heat equation (1.7) are both linear equations and deal with the evolution of a single scalar quantity. As nature is too complicated to be described by scalar linear equations, their utility is limited. Next, we present a *nonlinear system* of conservation laws.

1.1.3. Euler equations of gas dynamics. A gas (as an example consider air) consists of a large number of molecules. The motion of each molecule can be tracked individually. This description is termed as the particle description and leads to a very large number of ODEs. The resulting system of ODEs is too large to be computationally feasible. Instead, a more *macroscopic description* is used. In a macroscopic model, the key variables of interest are: the density ρ, the velocity field $\mathbf{u}$ and the gas pressure p. All these quantities can be measured experimentally. The relevant conservation laws are

- *Conservation of mass:* It is well-known in fluid dynamics that the total mass of the gas is conserved. Mathematically, using Kelvin's theorem, this translates into
$$\rho_t + \operatorname{div}(\rho \mathbf{u}) = 0.$$
- *Conservation of momentum:* By Newton's second law of motion, the rate of change of momentum equals force. In the absence of external forces, the gas pressure is the only force acting on the gas. The resulting conservation law is
$$(\rho \mathbf{u})_t + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = 0.$$
Note that the above conservation laws implies that the rate of change of the *advective* (material) derivative of the momentum equals the gradient of pressure. This is a consequence of the observation that gas flows from high to low pressure. In the above equation, the symbol $\otimes$ is the *tensor product*
$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{pmatrix}$$
for any two vectors $\mathbf{a} = (a_1, a_2, a_3)$ and $\mathbf{b} = (b_1, b_2, b_3)$.
- *Conservation of energy:* The total energy of a gas is a sum of its kinetic and internal (potential) energy. The kinetic energy has the standard expression
$$E_k = \frac{1}{2}\rho |\mathbf{u}|^2,$$
whereas the internal energy is determined by an equation of state. If the gas is an ideal gas, then the equation of state is
$$E_i = \frac{p}{\gamma - 1},$$
where γ is the gas constant. It takes the values $5/3$ and $7/5$ for mono-atomic and diatomic gases, respectively. Hence, the total energy of an ideal gas is
$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho |\mathbf{u}|^2. \tag{1.9}$$
The rate of change of total energy is computed as:
$$E_t + \operatorname{div}((E + p)\mathbf{u}) = 0.$$

All the three conservation laws are combined together and written in divergence form to obtain the *Euler equations* of gas dynamics:

$$\begin{aligned} \rho_t + \operatorname{div}(\rho \mathbf{u}) &= 0, \\ (\rho \mathbf{u})_t + \operatorname{div}(\rho \mathbf{u} \otimes \mathbf{u} + p\mathbf{I}) &= 0, \\ E_t + \operatorname{div}((E + p)\mathbf{u}) &= 0, \end{aligned} \tag{1.10}$$

where $\mathbf{I}$ denotes the 3×3 identity matrix. The above system is an example of a multi-dimensional *nonlinear system* of conservation laws. This derivation of the Euler equations was very brief and details can be found in fluid dynamics textbooks like [**LL87**]. We ignore fluid viscosity effects and heat conduction in the gas while deriving (1.10).

The above examples already reveal a multitude of diverse physical phenomena that can be modeled in terms of conservation laws. The flux $\mathbf{F}$ in (1.4) is often a function of $\mathbf{U}$ and its derivatives,
$$\mathbf{F} = \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}, \nabla^2 \mathbf{U}, \dots)$$

For simplicity of the analysis, it is common to neglect the role of the higher than first-order derivatives. Hence, the flux is of the form:

$$\mathbf{F} = \mathbf{F}(\mathbf{U}, \nabla\mathbf{U}).$$

If it is of the form $\mathbf{F} = \mathbf{F}(\mathbf{U})$, then the conservation law (1.4) is a first-order PDE. It is usually classified as *hyperbolic*. The notion of hyperbolicity will be described in detail in the sequel. The scalar transport equation (1.5) and the Euler equations of gas dynamics (1.10) are examples for hyperbolic equations.

If we have $\mathbf{F} = \mathbf{F}(\nabla\mathbf{U})$, then the conservation law (1.4) is a second-order PDE and is often classified as *parabolic*. The heat equation (1.7) is an example of a parabolic equation. When the flux $\mathbf{F}$ depends on both the function $\mathbf{U}$ and its first derivative, the conservation law (1.4) is termed as a *convection-diffusion* equation. In these notes, we will consider hyperbolic equations and convection-diffusion equations with the convection dominating the diffusion.

1.1.4. Other examples. Examples for conservation laws of both the hyperbolic and convection-diffusion type abound in nature. In these notes, we will consider the scalar Burgers equation, the Buckley-Leverett equation (modeling flows in oil and gas reservoirs), the wave equation, the shallow water equations of meteorology and oceanography, the equations for linear and nonlinear elastic waves that arise in materials science and the equations of magnetohydrodynamics (MHD) from plasma physics.

1.2. Content and scope of these notes

The reason for studying conservation laws extensively is obvious: They arise in many models in the sciences, ranging from the design of aircraft (Euler equations) to the study of supernovas in astrophysics (MHD equations). Since interesting conservation laws like the Euler equations are nonlinear, it is not possible to obtain explicit solution formulas. Hence, numerical methods need to be developed for *approximating* or *simulating* the solutions of conservation laws. The design and implementation of efficient numerical methods is the main focus of these notes.

In order to design efficient numerical methods, we need to understand the analytical structure of the solutions of conservation laws. Therefore, we will briefly discuss theoretical properties of the solutions that are relevant for the design and analysis of numerical schemes.

We begin with the study of one-dimensional scalar problems. Both linear and nonlinear equations are considered, and efficient numerical schemes are described for them. Then, the focus shifts to linear and nonlinear systems like the Euler equations of gas dynamics. Finally, we consider the multi-dimensional versions of systems of conservation laws and describe efficient numerical schemes for them.

CHAPTER 2

Linear Transport Equations

In this chapter we consider the one-dimensional version of the linear transport equation,

$$U_t + a(x,t)U_x = 0 \qquad \forall\ (x,t) \in \mathbb{R} \times \mathbb{R}_+. \tag{2.1}$$

The simplest case of the scalar transport equation arises when the velocity field is constant, that is, $a(x,t) \equiv a$. The resulting transport equation is

$$U_t + aU_x = 0. \tag{2.2}$$

The rather simple equation (2.2) has served as a crucible for designing highly efficient schemes for much more complicated systems of equations. We concentrate on it for the rest of this chapter.

2.1. Method of characteristics

The initial value problem (or Cauchy problem) for (2.1) consists of finding a solution of (2.1) that also satisfies the initial condition

$$U(x,0) = U_0(x) \qquad \forall\ x \in \mathbb{R}. \tag{2.3}$$

It is well known that the solution of the initial value problem can be constructed by using the *method of characteristics*. The idea underlying this method is to reduce a PDE like (2.1) to an ODE by utilizing the structure of the equation. As an ansatz, assume that we are given some curve $x(t)$, along which the solution U is constant. This means that

$$\begin{aligned} 0 &= \frac{d}{dt}U(x(t),t) && \text{(as U is constant along } x(t)) \\ &= U_t(x(t),t) + U_x(x(t),t)x'(t) && \text{(chain rule).} \end{aligned}$$

We also know that $U_t(x(t),t) + U_x(x(t),t)a(x(t),t) = 0$, since U is assumed to be a solution of (2.1). Therefore, if $x(t)$ satisfies the ODE

$$\begin{aligned} x'(t) &= a(x(t),t) \\ x(0) &= x_0, \end{aligned} \tag{2.4}$$

then $x(t)$ is precisely such a curve. The solution $x(t)$ of this equation is called a *characteristic curve*. From ODE theory, we know that solutions of (2.4) exist provided that a is Lipschitz continuous in both arguments. It may or may not be possible to find an explicit solution formula for (2.4).

The importance of characteristic curves lies in the property that U is constant along them:

$$U(x(t),t) = U(x(0),0) = U_0(x_0).$$

The initial data $U_0(x)$ is already known, so if we can find characteristic curves that go through all points $(x,t) \in \mathbb{R} \times \mathbb{R}_+$, then we have found the solution U at all points in the plane. (See Figure 2.1) for an illustration.)

In the simple case of a constant velocity field $a(x,t) \equiv a$, the characteristic equation (2.4) is explicitly solved as

$$x(t) = x_0 + at.$$

Therefore, given some point (x,t), the unique characteristic that goes through (x,t) (so that $x(t) = x$) has initial value $x_0 = x - at$. Hence, the solution of (2.2) is

$$U(x,t) = U_0(x_0) = U_0(x - at) \tag{2.5}$$

for any $(x,t) \in \mathbb{R} \times R_+$. The solution formula (2.5) implies that the initial data is transported with the velocity a.

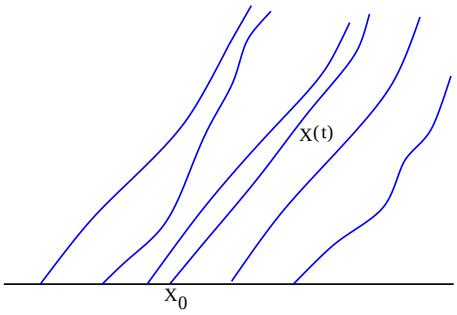


FIGURE 2.1. Characteristics curves $x(t)$ for (2.1)

In the more general case of (2.1), the characteristic equation (2.4) may not be possible to solve explicitly. Hence, it is essential that we obtain some information about the structure of solutions of (2.1) from the equation itself. This is done by means of the following *a priori* energy estimate:

Lemma 2.1. *Let $U(x,t)$ be a smooth solution of (2.1) which decays to zero at infinity, i.e, $\lim_{|x|\to\infty} U(x,t) = 0$ for all $t \in \mathbb{R}_+$, and assume that $a \in C^1(\mathbb{R}, \mathbb{R}_+)$. Then U satisfies the energy bound*

$$\int_{\mathbb{R}} U^2(x,t)\,dx \leqslant e^{\|a\|_{C^1} t} \int_{\mathbb{R}} U_0^2(x)\,dx \tag{2.6}$$

for all times $t > 0$.

Proof. The proof of the estimate (2.6) is based on multiplying (2.1) with U on both sides:

$$\begin{aligned}
UU_t + a(x,t)\,UU_x &= 0 && \text{(multiplying (2.1) by } U\text{)}\\
\left(\frac{U^2}{2}\right)_t + a(x,t)\left(\frac{U^2}{2}\right)_x &= 0 && \text{(chain rule)}\\
\left(\frac{U^2}{2}\right)_t + \left(a(x,t)\frac{U^2}{2}\right)_x &= a_x(x,t)\frac{U^2}{2} && \text{(product rule)}\\
\frac{d}{dt}\int_{\mathbb{R}}\left(\frac{U^2}{2}\right)dx + \int_{\mathbb{R}}\left(a(x,t)\frac{U^2}{2}\right)_x dx &= \int_{\mathbb{R}} a_x(x,t)\frac{U^2}{2}\,dx && \text{(integrating over space)}\\
\frac{d}{dt}\int_{\mathbb{R}}\left(\frac{U^2}{2}\right)dx &= \int_{\mathbb{R}} a_x(x,t)\frac{U^2}{2}\,dx && \text{(decay to zero at infinity)}\\
&\leqslant \|a\|_{C^1}\int_{\mathbb{R}}\frac{U^2}{2}\,dx && \text{(regularity of } a\text{).}
\end{aligned}$$

The last inequality can be used together with Gronwall's inequality (Theorem A.1) to obtain the bound (2.6). □

The quantity $\int U^2/2$ is commonly called the *energy* of the solution. The above lemma shows that the energy of the solutions to the transport equation (2.1) are bounded. The energy estimate is going to be used for designing robust schemes for the transport equation. We remark that the restriction that U decays to zero at infinity may be relaxed by considering a different energy functional.

The solution is also bounded in L^∞:

Lemma 2.2. *If U is a smooth solution of (2.1) and $U_0 \in L^\infty(\mathbb{R})$, then for any $t > 0$, $\sup_{x\in\mathbb{R}} |U(x,t)| \leqslant \|U_0\|_{L^\infty}$.*

Proof. We know that for any $x \in R$ and $t \in \mathbb{R}^+$, there exists $\xi \in \mathbb{R}$ such that $U(x,t) = U_0(\xi)$. This shows that $|U(x,t)| \leqslant \|U_0\|_{L^\infty}$ for all $x \in \mathbb{R}$. □

2.2. Finite difference schemes for the transport equation

It may not be possible to obtain an explicit formula for the solution of the characteristic equation (2.4). For example, the velocity field $a(x,t)$ might have a complicated nonlinear expression. Hence, we

have to devise numerical methods for approximating the solutions of (2.1). For simplicity, we consider $a(x,t) \equiv a > 0$ and solve (2.2). It is rather straightforward to extend the schemes to the case of a more general velocity field.

2.2.1. Discretization of the domain. The first step in any numerical method is to discretize both the spatial and temporal parts of the domain. Since $\mathbb{R}$ is unbounded, we have to truncate the domain to some bounded domain $[x_L, x_R]$. This truncation implies that suitable boundary conditions need to be imposed. We discuss the problem of boundary conditions later on.

For the sake of simplicity, the domain $[x_L, x_R]$ is discretized uniformly with a mesh size Δx into a sequence of $N+1$ points x_j such that $x_0 = x_L$, $x_N = x_R$ and $x_{j+1} - x_j = \Delta x$ for all j. A non-uniform discretization can readily be considered.

For the temporal discretization, we choose some terminal time T and divide $[0,T]$ into M points $t^n = n\Delta t$ $(n = 0, \dots, M)$. The space-time mesh is shown in Figure 2.2. Our aim is obtain an approximation of the form $U_j^n \approx U(x_j, t^n)$. To get from the initial time step t^0 to the terminal time step t^M, we first set the initial data $U_j^0 = U_0(x_0)$ for all j. Then the solution U_j^1 at the next time step is computed using some update formula, again for all j. This process is reiterated until we arrive at the final time step $t^M = T$ with our final solution U_j^M.

2.2.2. A simple centered finite difference scheme. On the mesh, we need to approximate the transport equation (2.2). We do so by replacing both the spatial and temporal derivatives by finite differences. The time derivative is replaced with a forward difference and the spatial derivative with a central difference. This combination is standard (see schemes for the heat equation in standard textbooks like [**TW09**]). The resulting scheme is

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = 0 \qquad \text{for } j = 1, \dots, N-1. \tag{2.7}$$

Some special care must be taken when defining the boundary values. We have a consistent discretization of (2.2) that is very simple to implement. We test it on the following numerical example.

2.2.3. A numerical example. Consider the linear transport equation (2.2) in the domain $[0,1]$ with initial data

$$U_0(x) = \sin(2\pi x). \tag{2.8}$$

Since the data is periodic, it is natural to assume periodic boundary conditions. We implement this numerically by letting

$$U_0^n = U_{N-1}^n, \quad U_N^n = U_1^n.$$

The exact solution is calculated by (2.5) as $U(x,t) = \sin(2\pi(x - at))$. We set $a = 1$ and compute the solutions with the central scheme (2.7) with 500 mesh points, and plot the solution at time $t = 0.3$ in Figure 2.3. The figure clearly shows that, despite being a consistent approximation, the scheme is unstable, with very large oscillations.

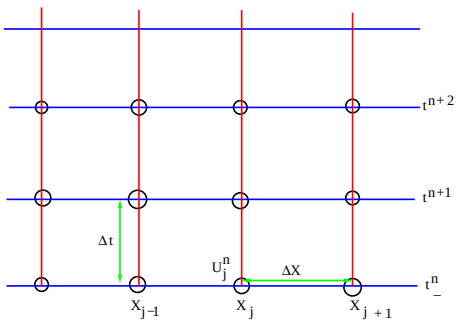


FIGURE 2.2. A representation of the mesh in space-time

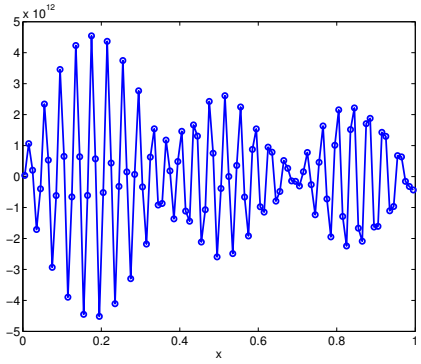


FIGURE 2.3. Approximate solution for (2.2) with the central scheme (2.7) at time $t = 3$ with 100 mesh points. `[central.m]`

2.2.4. A physical explanation. Why do the solutions computed with the central scheme (2.7) blow up? After all, the central scheme seems a reasonable approximation of the transport equation. A *physical* explanation can be deduced from the following argument: The exact solution moves to the right (as $a > 0$) with a fixed speed. Therefore, information goes from left to right. However, the central scheme (see Figure 2.4) takes information from both the left and the right, violating the physics. Consequently, the solutions are unstable. This explanation seems intuitive but has to be backed by solid mathematical arguments. We proceed to do so below.

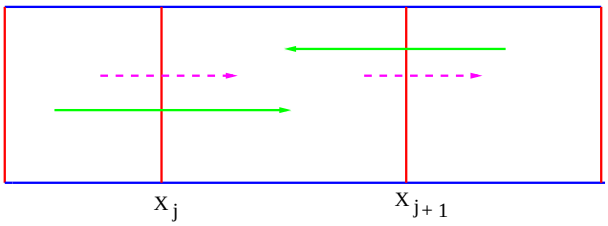


FIGURE 2.4. The central scheme (2.7). Green arrows indicate numerical propagation and magenta arrows physical propagation.

2.2.5. A mathematical explanation. The observed instability of the central scheme can be explained mathematically in terms of estimates. We recall that the exact solutions have a bounded energy (see estimate (2.6)). It is reasonable to require that the scheme is *energy stable* like the exact solution, that is, a discrete version of energy remains bounded. For a given Δx, we define the discrete version of energy as

$$E^n = \frac{1}{2}\Delta x \sum_j \left(U_j^n\right)^2. \tag{2.9}$$

Note that the integral in the energy for the continuous problem has been replaced with a Riemann sum.

Lemma 2.3. *Let U_j^n be the solutions computed with the central scheme* (2.7)*. Then the following estimate holds:*

$$E^{n+1} = E^n + \frac{\Delta x}{2} \sum_j \left(U_j^{n+1} - U_j^n\right)^2. \tag{2.10}$$

Consequently, the energy grows at every time step for any choice of $\Delta x, \Delta t$.

Proof. We mimic the steps of continuous energy estimate (Lemma 2.1) and multiply both sides of the scheme (2.7) by U_j^n to obtain

$$U_j^n\left(U_j^{n+1} - U_j^n\right) + \frac{a\Delta t}{2\Delta x}\left(U_j^n U_{j+1}^n - U_j^n U_{j-1}^n\right) = 0. \tag{2.11}$$

We have the following elementary identity:

$$d_2(d_1 - d_2) = \frac{(d_1)^2}{2} - \frac{(d_2)^2}{2} - \frac{1}{2}(d_1 - d_2)^2 \tag{2.12}$$

for any two numbers d_1, d_2. We denote

$$H_{j+1/2} = a\frac{U_j^n U_{j+1}^n}{2}$$

to reduce (2.11) to

$$\frac{\left(U_j^{n+1}\right)^2}{2} = \frac{(U_j^n)^2}{2} + \frac{1}{2}\left(U_j^{n+1} - U_j^n\right)^2 - \frac{\Delta t}{\Delta x}(H_{j+1/2} - H_{j-1/2}). \tag{2.13}$$

Summing (2.13) over all j and using zero (or periodic) boundary conditions, the flux term H vanishes by cancellation and we obtain the estimate (2.10). □

Although we assumed zero or periodic boundary conditions in the proof of this lemma, a variant of the lemma holds for more general boundary conditions, as for the continuous setting in Lemma 2.1.

The above lemma provides a mathematical justification for our physical intuition. The central scheme leads to a growth of energy at every time step and is unstable. We need to find schemes that posses a discrete version of the energy estimate. This use of rigorous mathematical tools like energy analysis to justify physical reasoning will be an essential ingredient of these notes.

2.3. An upwind scheme

The central scheme (2.7) does not respect the direction of propagation of information for the transport equation (2.2). Hence, we must include the correct direction of information propagation and hope that it stabilizes the scheme. This entails using one-sided differences instead of a central difference to approximate the linear transport equation (2.2).

If $a > 0$ and the direction of information propagation is from left to right, then we can use a backward difference in space to obtain the scheme

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_j^n - U_{j-1}^n)}{\Delta x} = 0 \qquad \text{for } j = 1, \ldots, N-1, \tag{2.14}$$

and if $a < 0$, we can use the forward difference to obtain:

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_j^n)}{\Delta x} = 0 \qquad \text{for } j = 1, \ldots, N-1. \tag{2.15}$$

Using the notation

$$a^+ = \max\{a, 0\}, \quad a^- = \min\{a, 0\}, \quad |a| = a^+ - a^-,$$

(2.14) and (2.15) can be written together as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a^+(U_j^n - U_{j-1}^n)}{\Delta x} + \frac{a^-(U_{j+1}^n - U_j^n)}{\Delta x} = 0. \tag{2.16}$$

The above scheme takes into account the direction of propagation of information – information is "carried with the wind". Hence, this scheme is termed as the *upwind* scheme.

Using the definition of the absolute value and some simple algebraic manipulations, the upwind scheme (2.16) can be recast as

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = \frac{|a|}{2\Delta x}(U_{j+1}^n - 2U_j^n + U_{j-1}^n) \tag{2.17}$$

(compare to (2.7)). Note that in the above form, the spatial derivatives are the central term and a *diffusion* term. The right hand side of (2.17) approximates $\frac{\Delta x|a|}{2}U_{xx}$. Hence, the upwind scheme (2.17) adds *numerical viscosity* or *diffusion* to the unstable central scheme (2.7). Numerical viscosity is going to play a crucial role later on.

Since the upwind scheme incorporates the correct direction of propagation of information (see Figure 2.5), we expect it to be more stable than the central scheme. This is endorsed by the numerical experiment with initial data (2.8). We take $a = 1$ and compute approximate solutions for the linear transport equation (2.2) on a uniform mesh with 100 mesh points up to $t = 1$. We use two different timesteps: $\Delta t = 1.3\Delta x$

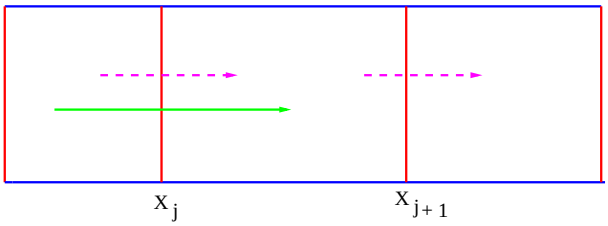
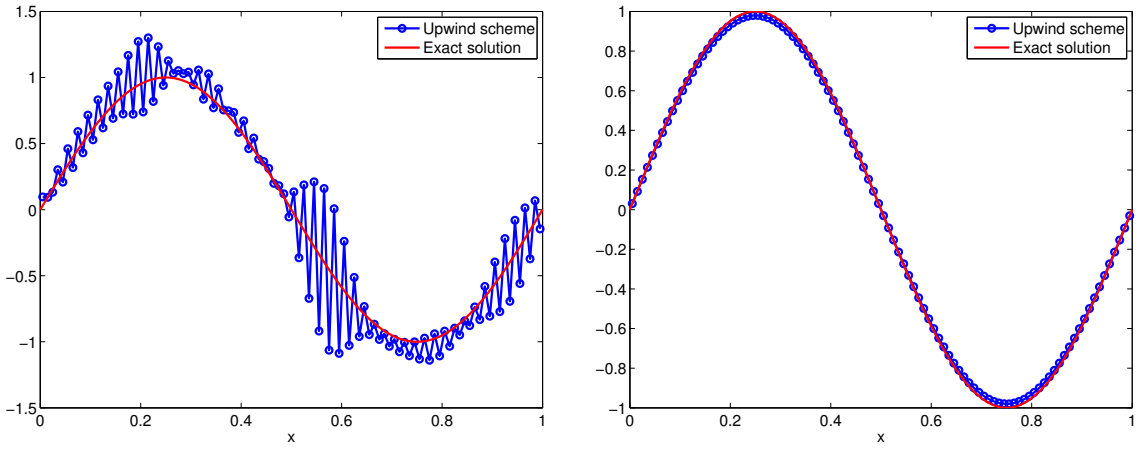


FIGURE 2.5. The upwind scheme (2.16). Green arrows indicate numerical propagation and magenta arrows physical propagation.



(a) $\Delta t = 1.3\Delta x$ (b) $\Delta t = 0.9\Delta x$

FIGURE 2.6. Solution with initial data (2.8) at $t = 1$. The ratio $\Delta t/\Delta x$ is important for stability. `[upwind_cfl.m]`

and $\Delta t = 0.9\Delta x$. As seen in Figure 2.6, the results with $\Delta t = 1.3\Delta x$ are still oscillatory and the scheme continues to be unstable. In spite of the upwinding, stability stills seems to be elusive. However, results with $\Delta t = 0.9\Delta x$ are stable. The approximation appears to be good in this case. Much better results are obtained by refining the mesh, while keeping the ratio $\Delta t/\Delta x$ fixed, as is presented in Figure 2.7.

2.4. Stability for the upwind scheme: L^1, L^2 and L^∞ norms

The numerical results indicate that stability for the upwind scheme is subtle. It is not unconditionally unstable as the central scheme (2.7); instead, stability depends on the parameters $\Delta x, \Delta t$. Numerical results indicate the crucial role played by the ratio $\frac{\Delta t}{\Delta x}$. It seems that one must not only take into account the correct direction of propagation, but also the correct magnitude.

The quantification of stability will involve energy analysis as in the last section. We have the following stability result:

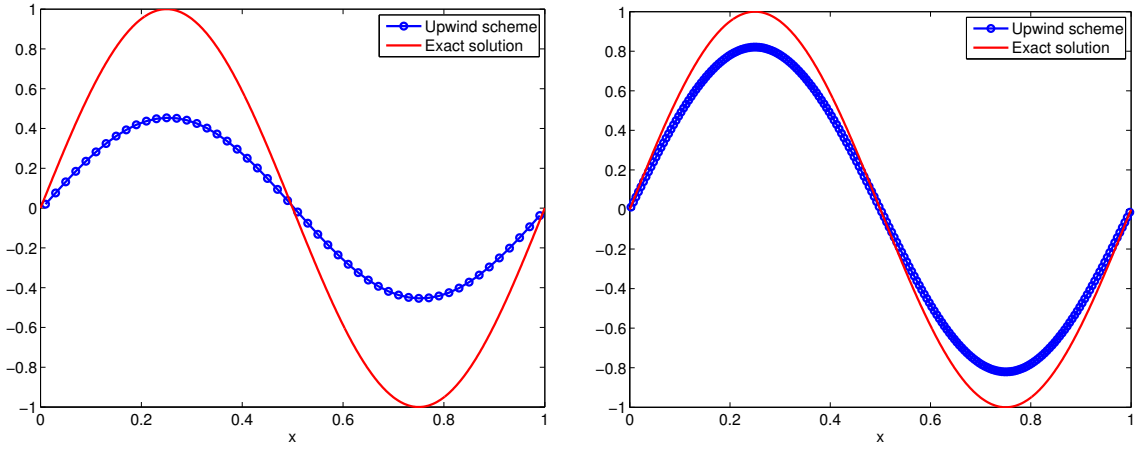
Lemma 2.4. *Let the mesh parameters satisfy the condition*

$$|a|\frac{\Delta t}{\Delta x} \leqslant 1. \tag{2.18}$$

Then solutions computed with the upwind scheme (2.17) *satisfy the energy estimate*

$$E^{n+1} \leqslant E^n, \tag{2.19}$$

where the energy is defined as in (2.9). *The upwind scheme is thus* conditionally stable.



(a) 50 mesh points (b) 200 mesh points

FIGURE 2.7. Solution with initial data (2.8) at $t = 10$. Refining the mesh gives a more accurate solution. `[upwind_refinement.m]`

Proof. For the sake of simplicity, we assume that $a > 0$. Hence the upwind scheme (2.17) reduces to

$$\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{a(U_{j+1}^n - U_{j-1}^n)}{2\Delta x} = \frac{a}{2\Delta x}(U_{j+1}^n - 2U_j^n + U_{j-1}^n). \tag{2.20}$$

It is also equivalent to the scheme (2.14). As in the proof of the estimate (2.10) we multiply both sides of the scheme (2.20) by U_j^n to obtain

$$\begin{aligned} U_j^n(U_j^{n+1} - U_j^n) = &-\frac{a\Delta t}{2\Delta x}(U_j^n U_{j+1}^n - U_j^n U_{j-1}^n) \\ &+ \frac{a\Delta t}{2\Delta x}(U_j^n(U_{j+1}^n - U_j^n)) + \frac{a\Delta t}{2\Delta x}(U_j^n(U_{j-1}^n - U_j^n)). \end{aligned} \tag{2.21}$$

Now we use elementary identity (2.12) a couple of times and rewrite (2.21) as

$$\begin{aligned} \frac{(U_j^{n+1})^2}{2} = &\frac{(U_j^n)^2}{2} + \frac{(U_j^{n+1} - U_j^n)^2}{2} - \frac{a\Delta t}{2\Delta x}(U_j^n U_{j+1}^n - U_j^n U_{j-1}^n) \\ &+ \frac{a\Delta t}{4\Delta x}\left((U_{j+1}^n)^2 - (U_j^n)^2\right) - \frac{a\Delta t}{4\Delta x}\left((U_j^n)^2 - (U_{j-1}^n)^2\right) \\ &- \frac{a\Delta t}{4\Delta x}(U_{j+1}^n - U_j^n)^2 - \frac{a\Delta t}{4\Delta x}(U_j^n - U_{j-1}^n)^2. \end{aligned} \tag{2.22}$$

Denoting

$$K_{j+1/2} = \frac{a}{2}(U_j^n U_{j+1}^n) - \frac{a}{4}\left((U_{j+1}^n)^2 - (U_j^n)^2\right),$$

we may rewrite (2.22) as

$$\begin{aligned} \frac{(U_j^{n+1})^2}{2} = &\frac{(U_j^n)^2}{2} + \frac{(U_j^{n+1} - U_j^n)^2}{2} - \frac{a\Delta t}{\Delta x}(K_{j+1/2} - K_{j-1/2}) \\ &- \frac{a\Delta t}{4\Delta x}(U_{j+1}^n - U_j^n)^2 - \frac{a\Delta t}{4\Delta x}(U_j^n - U_{j-1}^n)^2. \end{aligned} \tag{2.23}$$

Summing (2.23) over all j and using the definition of discrete energy (2.9) and either zero or periodic boundary conditions, we obtain

$$E^{n+1} \leqslant E^n + \frac{\Delta x}{2}\sum_j (U_j^{n+1} - U_j^n)^2 - \frac{a\Delta t}{2}\sum_j (U_j^n - U_{j-1}^n)^2. \tag{2.24}$$

Using the definition of the upwind scheme (2.14) in (2.24) yields

$$E^{n+1} \leqslant E^n + \left(\frac{a^2\Delta t^2}{2\Delta x} - \frac{a\Delta t}{2}\right)\sum_j (U_j^n - U_{j-1}^n)^2. \tag{2.25}$$

Since the term in the sum in (2.25) is positive, we obtain the energy bound (2.19), provided

$$\frac{a^2\Delta t^2}{\Delta x} \leqslant a\Delta t,$$

which is precisely the condition (2.18). □

The stability condition (2.18) is termed the *CFL condition* after Courant, Friedrichs and Lewy who first proposed it. By a slightly different approach we can also show stability in L^1 and L^∞, as follows:

Lemma 2.5. *Assume that the CFL condition* (2.18) *is satisfied. Then solutions computed with the upwind scheme* (2.17) *satisfy the stability estimates*

$$\|U^{n+1}\|_{L^1} \leqslant \|U^n\|_{L^1}, \qquad \|U^{n+1}\|_{L^\infty} \leqslant \|U^n\|_{L^\infty} \qquad \forall\ n = 0, 1, 2, \ldots, \tag{2.26}$$

where $\|U\|_{L^1} = \Delta x \sum_{j\in\mathbb{Z}} |U_j|$ *and* $\|U\|_{L^\infty} = \sup_{j\in\mathbb{N}} |U_j|$.

Proof. Denoting $\nu = \frac{\Delta t}{\Delta x}$, we can rewrite (2.17) as

$$\begin{aligned} U_j^{n+1} &= U_{j+1}^n \left(\frac{\nu|a| - \nu a}{2}\right) + U_j^n\,(1 - \nu|a|) + U_{j-1}^n \left(\frac{\nu|a| + \nu a}{2}\right) \\ &= U_{j+1}^n\left(-\nu a^-\right) + U_j^n\,(1 - \nu|a|) + U_{j-1}^n \nu a^+. \end{aligned}$$

Thus, the CFL guarantees that U_j^{n+1} is a convex combination of U_{j-1}^n, U_j^n and U_{j+1}^n (i.e., a linear combination with positive coefficients which sum up to 1). This ensures that $|U_j^{n+1}| \leqslant \max(|U_{j-1}^n|, |U_j^n|, |U_{j+1}^n|)$. For the L^1 bound, take the absolute value of the above and sum over $j \in \mathbb{N}$:

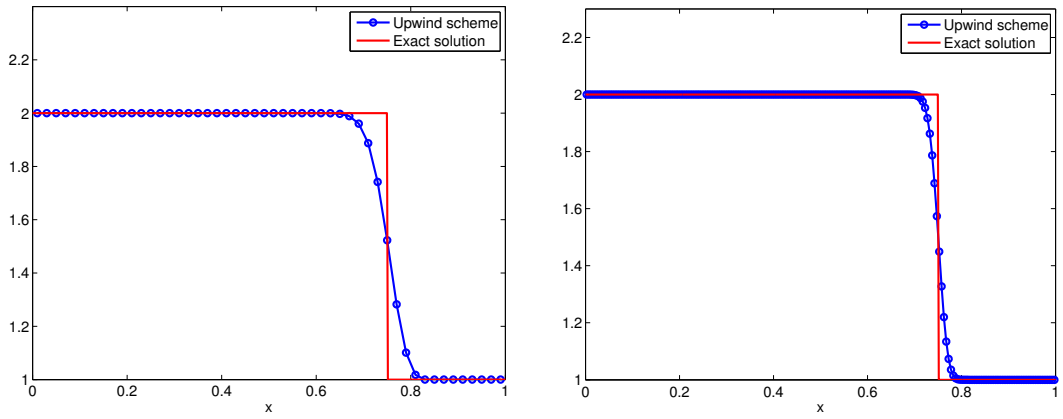
$$\begin{aligned} \sum_j |U_j^{n+1}| &= \sum_j \left|U_{j+1}^n\left(-\nu a^-\right) + U_j^n\,(1 - \nu|a|) + U_{j-1}^n \nu a^+\right| \\ &\leqslant \sum_j |U_{j+1}^n|\left(-\nu a^-\right) + \sum_j |U_j^n|\,(1 - \nu|a|) + \sum_j |U_{j-1}^n|\nu a^+ \\ &= \sum_j |U_j^n|\left((-\nu a^-) + (1 - \nu|a|) + \nu a^+\right) \\ &= \sum_j |U_j^n|. \end{aligned}$$

□

Numerical experiment: Discontinuous data. Consider the transport equation (2.2) with $a = 1$ in the domain $[0, 1]$ and initial data

$$U_0(x) = \begin{cases} 2 & \text{if } x < 0.5 \\ 1 & \text{if } x > 0.5. \end{cases} \tag{2.27}$$

The initial data and consequently the exact solution (2.5) are discontinuous. We compute with the upwind scheme using 50 and 200 mesh points and display the results in Figure 2.8. The results show that the upwind scheme approximates the solution quite well, at least at a fine resolution. However the errors on a coarse mesh are somewhat large. This issue will be addressed in later sections.



(a) Upwind 50 mesh points

(b) Upwind 200 mesh points

FIGURE 2.8. The upwind scheme (2.14) for the linear advection equation (2.2) with discontinuous initial data (2.27). Results are at time $t = 0.25$. `[upwind_disc_refinement.m]`

CHAPTER 3

Scalar conservation laws

In the previous chapter, we considered the scalar transport equation

$$U_t + a(x,t)U_x = 0. \tag{3.1}$$

This equation is linear as the velocity field a is a given function. However, most natural phenomena are nonlinear. In such models, the velocity field depends on the solution itself. The simplest example of such a field is

$$a(x,t) = U(x,t).$$

Hence, the transport equation (3.1) becomes

$$U_t + UU_x = 0. \tag{3.2}$$

The transport equation (3.2) can be written in the *conservative form*

$$U_t + \left(\frac{U^2}{2}\right)_x = 0. \tag{3.3}$$

This is the *inviscid Burgers equation*. It serves as a prototype for *scalar conservation laws*, which in general take the form

$$U_t + f(U)_x = 0, \tag{3.4}$$

where U is the unknown and f is the flux function. Apart from Burgers' equation, scalar conservation laws arise in a wide variety of models. We consider a couple of examples below.

Traffic flow model. For simplicity, consider a one-dimensional highway and denote the density of cars (number of cars per square meter) as $U(x,t)$. Assume that the cars are moving at a *macroscopic* velocity (the speed of a traffic column) $V(x,t)$. A simple requirement of conservation of the number of cars lead to the following equation:

$$U_t + (UV)_x = 0. \tag{3.5}$$

The velocity V remains to be modeled. One very simple model is based on a couple of observations. First, there exists a maximum velocity at which an individual car can drive, for example specified by the speed limit. Second, the velocity of cars is inversely proportional to the car density. If there are a large number of cars, each individual driver will drive slowly. However, on a remote stretch of the highway, each driver speeds up. These simple observations are combined to yield the velocity

$$V = V_{\max}(1-U),$$

where $V_{\max}$ is the maximum velocity for the cars. We use the convention that the maximum density or *road carrying capacity* is 1. Hence, the traffic flow equation is

$$U_t + \big(V_{\max}U(1-U)\big)_x = 0. \tag{3.6}$$

Enhanced oil recovery. Oil is generally found in sub-surface reservoirs, inside permeable rocks. The primary stage of oil recovery consists of drilling into the rocks and extracting oil by applying pressure. Only 20 to 30 percent of the available oil can be extracted in this manner. The secondary stage of oil recovery consists of injecting water into the rock bed. The water displaces the oil (as water is heavier) and the oil can then be extracted. This complex process is modeled by using two-phase flow (water and oil) in a porous media (rock).

For simplicity, we assume that the reservoir is one-dimensional. The quantities of interest are the oil and water volume fractions or *saturations* S^o and S^w, respectively. Being volume fractions, they satisfy

$$S^o + S^w \equiv 1. \tag{3.7}$$

Furthermore, the phases evolve according to the conservation laws

$$\begin{cases} S^o_t + V^o_x = 0 \\ S^w_t + V^w_x = 0. \end{cases} \tag{3.8}$$

The phase velocities V^o, V^w are modeled by Darcy's law:

$$\begin{cases} V^o = -\lambda^o \frac{dP^o}{dx} \\ V^w = -\lambda^w \frac{dP^w}{dx}, \end{cases} \tag{3.9}$$

where λ and P are the phase mobility and the phase pressure, respectively. In the above constitutive relation, we have neglected the role of gravity. Furthermore, we can assume that there is no capillary pressure:

$$P^o = P^w.$$

Adding the phase saturation equations (3.8) for each phase and using the requirement (3.7), we obtain

$$(V^o + V^w)_x \equiv 0 \qquad \Rightarrow \qquad V^o + V^w = q,$$

for some constant q called the total flow rate. Substituting Darcy's law (3.9) in the above identity and using $P^w = P^o = P$, we obtain

$$\frac{dP}{dx} = -\frac{q}{\lambda^o + \lambda^w}.$$

Applying this identity in the evolution of the oil saturation (3.9) and (3.8) yields

$$S^o_t + \left(\frac{q\lambda^o}{\lambda^w + \lambda^o} \right)_x = 0. \tag{3.10}$$

The mobilities generally take the form

$$\lambda^o = (S^o)^2, \qquad \lambda^w = (S^w)^2 = (1 - S^o)^2.$$

Hence, the evolution of the oil saturation is governed by the scalar conservation law

$$S^o_t + \left(\frac{q(S^o)^2}{(S^o)^2 + (1 - S^o)^2} \right)_x = 0. \tag{3.11}$$

The above examples demonstrate that scalar conservation laws do occur in many interesting models in physics and engineering. Furthermore, the shape of the flux function f in (3.4) can be very general. Note that it is convex for Burgers' equation, concave for the traffic flow problem (3.6) and is neither convex nor concave (contains inflection points) for the oil reservoir equation (3.11).

In this section, we embark on a systematic study of scalar conservation laws (3.4) from a theoretical perspective.

3.1. Characteristics for Burgers' equation

We start with Burgers' equation (3.3) and attempt to construct solutions to the initial value problem associated with it. As for the linear transport equation (3.1), we will use the method of characteristics for this purpose. Since (3.2) and (3.3) are equivalent whenever U is smooth, the characteristics $x(t)$ for Burgers' equation are given by

$$\begin{aligned} x'(t) &= U(x(t), t) \\ x(0) &= x_0. \end{aligned} \tag{3.12}$$

Note that these characteristics are different from the linear case (2.4) in that the velocity depends on the solution. We start by considering the initial data

$$U_0(x) = \begin{cases} U_L & \text{if } x < 0 \\ U_R & \text{if } x > 0. \end{cases} \tag{3.13}$$

Data of this form is quite simple and consists of constants separated by a discontinuity at the origin. The initial value problem for a conservation law (3.4) with initial data of the form (3.13) is called a *Riemann problem*.

By definition, the solution U is constant along characteristics, that is, $U(x(t),t) = U_0(x_0)$. Therefore, the solution of (3.12), (3.13) in constant parts of U_0 is

$$x(t) = U_0(x_0)t + x_0.$$

Let $U_L = 1$ and $U_R = 0$ in (3.13). For $x_0 < 0$ the characteristics have velocity $U_0(x_0) = 1$, whereas for $x_0 > 0$ they have velocity 0; see Figure 3.1. We see that the characteristics intersect almost instantaneously. As observed in the last section, the solution should be constant (in time) along the characteristics. What happens to the solution when the characteristics start to intersect? How can the solution be defined in this case? Adding nonlinearity completely changes the situation from the linear case.

Is the intersection of characteristics on account of discontinuous data (3.13)? Can using smooth data lead to non-intersecting characteristics? It turns out that even smooth initial data can lead to the intersection of characteristics after a small time interval. Consider the visual example in Figure 3.2.

Exercise 3.1. *Let $U_0(x)$ be differentiable with at least one point x such that $U_0'(x) < 0$. Show that the solution to Burgers' equation with initial data U_0 will develop a discontinuity at time*

$$t_{\min} = -\frac{1}{\min_{x\in\mathbb{R}} U_0'(x)}.$$

(Hint: Start with the ansatz that two characteristics $x(t)$ and $\tilde{x}(t)$ intersect at some time t.)

The strange behavior of characteristics indicates that smooth solutions cannot be obtained for the conservation law (3.4), even when the initial data is smooth. Consider the initial data

$$U_0(x) = \sin(\pi x)$$

in the interval $[-1, 1]$. A heuristic interpretation of the characteristic equation (3.12) is that the solution at each point x moves with the velocity $U_0(x)$. Hence, the method of characteristics imply that the solution behaves as shown in Figure 3.3. The wave compresses in one part and stretches in another. In particular, the solution can be multi-valued. This is another indication that smooth solutions of (3.4) do not exist.

A formal calculation by differentiating (3.2) with respect to x yields

$$V_t + UV_x = -V^2, \tag{3.14}$$

where $V = U_x$. Hence, along the characteristics $x(t)$ given by (3.12), V varies as

$$\frac{d}{dt}V(x(t),t) = -V^2(x(t),t).$$

This is a ODE with quadratic nonlinearity and it is well known that the resulting solution V can blow up in finite time. Hence, the spatial derivative of the solution to Burgers' equation can blow up, even if the initial derivative is very small. This derivative blowup suggests that smooth solutions to (3.4) may not exist.

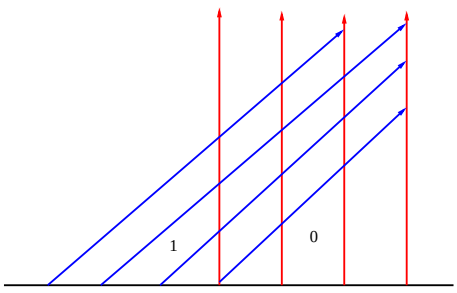
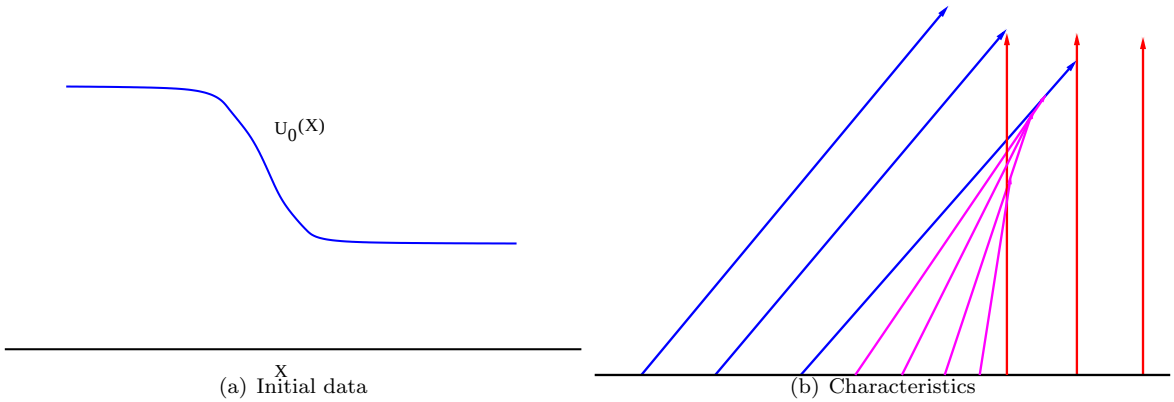


FIGURE 3.1. Characterstics intersecting for the Riemann problem (3.13) with $(U_L, U_R) = (1, 0)$.



(a) Initial data (b) Characteristics

FIGURE 3.2. Characteristics can even intersect for smooth initial data.

3.2. Weak solutions

The previous section demonstrates that smooth or classical solutions of the conservation law (3.4) may not exist. However, these models arise in physics and so *some* form of solution must exist. This type of solution is a *weak solution*. To motivate the definition of weak solutions, assume for the moment that smooth solutions of (3.4) exist and multiply both sides by a smooth test function $\varphi \in C_c^1(\mathbb{R} \times \mathbb{R}_+)$. (The space $C_c^1(A)$ is the space of all continuously differentiable functions from A to $\mathbb{R}$ with compact support, that is, the functions vanish outside a compact subset of A.) Integrating over $x \in \mathbb{R}$ and $t \in \mathbb{R}_+$ and integrating by parts, we find that

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}} U\varphi_t + f(U)\varphi_x \, dx \, dt + \int_{\mathbb{R}} U_0(x)\varphi(x,0) \, dx = 0. \tag{3.15}$$

This identity hold trues for all test functions $\varphi \in C_c^1(\mathbb{R} \times \mathbb{R}_+)$. We base the definition of weak solution for (3.4) on the above identity.

Definition 3.2 (Weak solution)**.** *A function $U \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ is a* weak solution *of* (3.4) *with initial data $U_0 \in L^\infty(\mathbb{R})$ if the identity* (3.15) *holds for all test functions $\varphi \in C_c^1(\mathbb{R} \times \mathbb{R}_+)$.*

Note that the identity (3.15) is well-defined as long as $U \in L^1_{\text{loc}}(\mathbb{R} \times \mathbb{R}_+)$.

Exercise 3.3. *Show that if a weak solution U of* (3.4) *is also differentiable (so $U \in C^1(\mathbb{R} \times \mathbb{R}_+)$), then U satisfies* (3.4) *pointwise. Hence, the class of weak solutions contains, but is not restricted to, classical solutions.*

Our usual understanding of solutions of PDEs is classical—the solutions must be differentiable functions. However, weak solutions are not necessarily differentiable, not even continuous. This implies that the solutions can contain discontinuities. These discontinuities appear in nature as *shock waves*.

3.2.1. The Rankine–Hugoniot condition. As we will soon find out, shock waves in weak solutions cannot be arbitrary curves in the x-t-plane, but must satisfy certain conditions. Assume that we

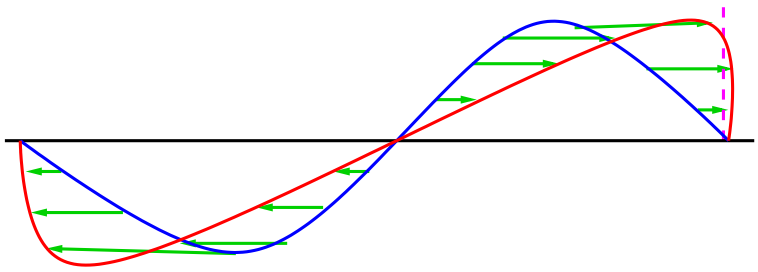


FIGURE 3.3. Smooth initial data leading to multi-valued solution.

are given a weak solution U consisting of two smooth regions, separated by a shock wave, as depicted in Figure 3.4. Let the shock wave be defined by the curve $x = \gamma(t)$.

Let φ be a test function with support in Ω, for some open set Ω which intersects the curve $x = \gamma(t)$; see Figure 3.4. We assume that $U \in C^1(\Omega^-)$ and $U \in C^1(\Omega^+)$. Integrating (3.15) by parts and using the compact support of the test function, we get

$$\begin{aligned}\int_\Omega U\varphi_t + f(U)\varphi_x \, d\Omega &= \int_{\Omega^+} U\varphi_t + f(U)\varphi_x \, d\Omega + \int_{\Omega^-} U\varphi_t + f(U)\varphi_x \, d\Omega \\ &= -\int_{\Omega^+} (U_t + f(U)_x)\varphi \, d\Omega + \int_{\partial\Omega^+} \big(U^+(t)\nu^t + f(U^+(t))\nu^x\big)\varphi \, d\Omega \\ &\quad -\int_{\Omega^-} (U_t + f(U)_x)\varphi \, d\Omega + \int_{\partial\Omega^-} \big(U^-(t)\nu^t + f(U^-(t))\nu^x\big)\varphi \, d\Omega \\ &= 0.\end{aligned}$$

Here, $U^+(t)$ and $U^-(t)$ are the trace values of U on the right and left of the discontinuity γ, and $\nu = (\nu^x, \nu^t)$ is the unit outward normal of γ (see Figure 3.4). Up to a normalization factor, the normal is

$$(\nu^x,\ \nu^t) = (1,\ -s(t)),$$

where $s(t) = \gamma'(t)$ is the speed of the shock curve. Since U is smooth in Ω^- and Ω^+, the equation (3.4) is satisfied pointwise. Therefore, the above identities imply that

$$\int_{\Omega^-\cup\Omega^+} \underbrace{(U_t + f(U)_x)}_{=\,0} \varphi \, d\Omega + \int_{\partial\Omega} \big(s(t)\,(U^+(t) - U^-(t)) - (f(U^+) - f(U^-))\big)\varphi \, d\Omega = 0.$$

Since φ is an arbitrary test function, the integrand of the remaining integral must be identically equal to zero. Hence, the shock speed must satisfy

$$s(t) = \frac{f(U^+(t)) - f(U^-(t))}{U^+(t) - U^-(t)}. \tag{3.16}$$

This condition is called the *Rankine–Hugoniot* condition. We summarize this as follows:

Theorem 3.4. *Let $\gamma \in C^1(\mathbb{R}_+)$ and let $U \in L^\infty(\mathbb{R}\times\mathbb{R}_+)$ be of the form*

$$U(x,t) = \begin{cases} U^-(x,t) & \text{if } x < \gamma(t) \\ U^+(x,t) & \text{if } x > \gamma(t) \end{cases}$$

where both U^- and U^+ are continuously differentiable functions. Then U is a weak solution of (3.4) if and only if *both*

- *U^- and U^+ solve* (3.4) *in the classical sense, and*
- *the shock speed $s(t) = \gamma'(t)$ satisfies the Rankine–Hugoniot condition* (3.16) *at $x = \gamma(t)$.*

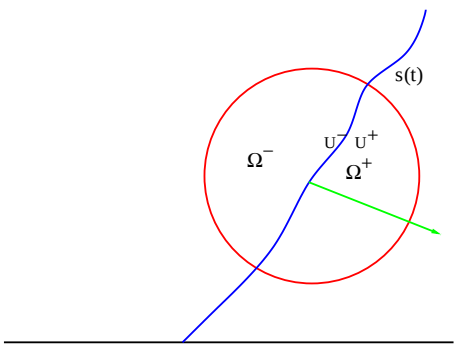


FIGURE 3.4. What happens across a shock?

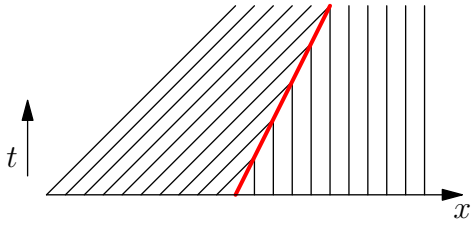


FIGURE 3.5. Characteristics for the Riemann problem (3.17).

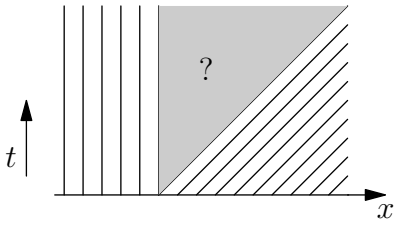


FIGURE 3.6. Characteristics for the Riemann problem (3.18).

3.2.2. Solutions to Riemann problems. Consider Burgers' equation (3.3) with the Riemann problem (3.13) with $U_L = 1$ and $U_R = 0$. We recall that the characteristics intersected in this case and a smooth solution couldn't be constructed. We construct a weak solution that consists of two constant states U_L and U_R, separated by a shock moving at a speed given by the Rankine–Hugoniot condition (3.16),

$$s(t) = \frac{\frac{U_R^2}{2} - \frac{U_L^2}{2}}{U_R - U_L} \equiv \frac{1}{2}.$$

Hence, the weak solution takes the form

$$U(x,t) = \begin{cases} 1 & \text{if } x < \frac{1}{2}t \\ 0 & \text{if } x > \frac{1}{2}t. \end{cases} \tag{3.17}$$

It is easy to check that (3.17) satisfies (3.15). The structure of the solution (see Figure 3.5) shows that the characteristics *flow into* the shock. As a consequence, there are characteristics covering all points in the plane, and for each point we can trace a characteristic back to the initial data. Hence, the entire solution is prescribed by the initial data.

Next, we consider another Riemann problem with $U_L = 0$ and $U_R = 1$. If we follow characteristics emanating from the x-axis, as for the previous problem, we now get an area without characteristics; see Figure 3.6. The "missing" information in this area may be "filled" in several ways. Using the Rankine–Hugoniot condition, we find that one possible weak solution is given by

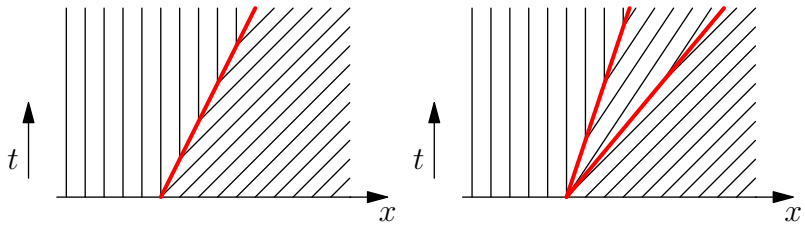
$$U(x,t) = \begin{cases} 0 & \text{if } x < \frac{1}{2}t \\ 1 & \text{if } x > \frac{1}{2}t, \end{cases} \tag{3.18}$$

see Figure 3.7(a). Note that this solution has one shock curve, drawn in red in the figure. However, this solution is not the only possible weak solution. By adding an intermediate state with value, say, $U_m = \frac{2}{3}$ and using the Rankine–Hugoniot condition, we get the weak solution

$$U(x,t) = \begin{cases} 0, & \text{if } x < \frac{1}{3}t \\ \frac{2}{3} & \text{if } \frac{1}{3}t < x < \frac{5}{6}t \\ 1 & \text{if } x > \frac{5}{6}t. \end{cases} \tag{3.19}$$

The characteristics are shown in Figure 3.7(b). In a similar manner one may construct arbitrarily many weak solutions by using the Rankine–Hugoniot condition (3.16) with different intermediate states.

This problem of non-uniqueness is implicit in the definition of weak solutions. These solutions are not necessarily unique, and therefore some extra conditions need to be imposed. For finding these extra



(a) Solution (3.18) (b) Solution (3.19)

FIGURE 3.7. Characteristics for different weak solutions the Riemann problem (3.18). Discontinuities are marked in red.

criteria, we observe that characteristics for both (3.18) and (3.19) *flow out* from the shock (see Figure 3.7). This is in contrast to the solution (3.17) where the characteristics flow *into* the shock (see Figure 3.5). Characteristics represent the flow of information. For an evolution equation the information should always flow from the initial data. This is clearly the case for the weak solution (3.17). However in the case of weak solutions (3.18) and (3.19), information seems to be created at the shock.

This heuristic requirement, that information is taken from the initial data and is not created at a shock, can be expressed in terms of conditions on the characteristics across a shock. Let $U^-(t)$, $U^+(t)$ be the states on either side of a shock with speed $s(t)$. The requirement that characteristics for Burgers' equation flow into the shock and information is taken from the initial line can be enforced by the condition

$$U^-(t) > s(t) > U^+(t). \tag{3.20}$$

It is simple to generalize (3.20) to the general scalar conservation law (3.4) for convex f:

$$f'(U^-(t)) > s(t) > f'(U^+(t)). \tag{3.21}$$

This is the *Lax entropy condition.*

Consider the conservation law (3.4) with a convex flux function and Riemann data (3.13). It is easily shown that

$$U(x,t) = \begin{cases} U_L & \text{if } x < st \\ U_R & \text{if } x > st, \end{cases} \tag{3.22}$$

where the shock speed s is defined by the Rankine–Hugoniot condition, is a weak solution of (3.4). Now, there are two cases: either $U_L > U_R$, or $U_L < U_R$. It turns out that the Lax entropy condition excludes (3.22) as a solution in the latter case, but not in the former:

Exercise 3.5. *Assume that f is strictly convex and that $U_L > U_R$. Show that* (3.22) *is a weak solution that satisfies the entropy condition* (3.21)*. Similarly, if $U_L < U_R$, show that* (3.22) *is a weak solution, but does* not *satisfy Lax' entropy condition.*

It turns out that in the latter case, where $U_L < U_R$, a continuous (but not necessarily differentiable) solution exists.

3.2.3. Rarefaction waves. For the remainder of this section, assume that the flux function f is strictly convex. In order to construct a continuous solution to (3.4), we note that replacing x, t by $\lambda x, \lambda t$ keeps the equation invariant, in the sense that a solution of one is a solution of the other. Since Riemann initial data (3.13) is also invariant with respect to the scaling $x \mapsto \lambda x$, it is natural to assume *self-similarity*—that solutions of the Riemann problem only depend on the ratio x/t:

$$U(x,t) = V\left(x/t\right). \tag{3.23}$$

Define the symmetry variable $\xi = x/t$. We substitute the ansatz (3.23) into (3.4) and use the chain rule repeatedly to obtain

$$\begin{aligned} 0 = U_t + f(U)_x &= V(\xi)_t + f'(V(\xi))V(\xi)_x \\ &= V'\xi_t + f'(V(\xi))V'\xi_x \\ &= -\frac{x}{t^2}V' + f'(V(\xi))\frac{1}{t}V' \end{aligned}$$

so

$$\big(f'(V(\xi)) - \xi\big)V' = 0.$$

In the nontrivial case of $V' \neq 0$, the above identity and the fact that f' is strictly increasing (recall that f is assumed to be strictly convex) leads to the expression

$$V(x/t) = (f')^{-1}(x/t). \tag{3.24}$$

A self-similar solution of this form is called a *rarefaction wave*.

The rarefaction wave can be employed to construct weak solutions for conservation laws. Consider the Riemann problem (3.4), (3.13). If $U_L < U_R$, then the weak solution is given by

$$U(x,t) = \begin{cases} U_L & \text{if } \ x \leqslant f'(U_L)t \\ (f')^{-1}(x/t) & \text{if } \ f'(U_L)t < x \leqslant f'(U_R)t \\ U_R & \text{if } \ x > f'(U_R)t. \end{cases} \tag{3.25}$$

Clearly (3.25) is a weak solution that satisfies Lax' entropy condition (3.21). For the particular case of Burgers' equation with Riemann data $U_L = 0$ and $U_R = 1$, the solution (3.25) is shown in Figure 3.8. Note how the characteristics are parallel to the rarefaction wave and contrast this to Figure 3.7.

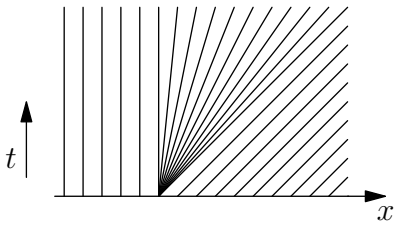


FIGURE 3.8. The rarefaction solution (3.25)

We now have a recipe to construct weak solutions for the Riemann problem (3.13) for a conservation law (3.4) with a strictly convex f. The solution depends on whether $U_L < U_R$ or $U_L > U_R$. If $U_L > U_R$, then the entropy satisfying weak solution (3.22) consists of a shock between the two states. If $U_L < U_R$, then the weak solution (3.25) consists of the two states, separated by a rarefaction wave. In both cases, the wave speed is bounded in absolute value by the maximum of $|f'(U_L)|$ and $|f'(U_R)|$.

We return to the solution of the Riemann problem for nonconvex fluxes in Section 3.4.

3.3. Entropy solutions

As we saw in Section 3.2.2, the Lax entropy condition (3.21) acts as a selection principle—it excludes certain weak solution, while admitting others. However, it is a *local* condition, referring only to the behavior of solutions at shocks, and therefore might be difficult to apply in a proof of global stability estimates. In this section we derive an alternative entropy condition which is equivalent to Lax' condition. As we will see, this condition guarantees stability and uniqueness of solutions to the Cauchy problem for the conservation law (3.4).

3.3.1. The entropy condition. A common technique in the study of PDEs is to add a viscous term to the PDE, study this new PDE, and then let the viscosity parameter go to zero. To this end we consider the following *viscous approximation* of the scalar conservation law (3.4):

$$U_t^\varepsilon + f(U^\varepsilon)_x = \varepsilon U_{xx}^\varepsilon, \tag{3.26}$$

where $\varepsilon > 0$ is a small parameter. The second-order term U_{xx} is termed the *viscous* or *diffusion* term, and adding this term turns the conservation law into a (parabolic) *convection-diffusion equation*. Such equations are similar to the heat equation (1.8), and as for the heat equation, the solutions to (3.26) are smooth, in fact C^∞ functions. See e.g. [**HR15**, Appendix B] or [**GR91**, Section II.2] for a rigorous study of (3.26).

Passing $\varepsilon \to 0$ in (3.26) formally gives back the scalar conservation law (3.4). A weak solution U which is the limit of solutions of the viscous equation, $U = \lim_{\varepsilon\to 0} U^\varepsilon$, is called a *vanishing viscosity solution* of (3.4). Instead of studying (3.26) in the limit $\varepsilon \to 0$, however, we derive some properties which such a limit would satisfy.

Let $\eta : \mathbb{R} \to \mathbb{R}$ be any strictly convex function, and construct the function $q : \mathbb{R} \to \mathbb{R}$,

$$q(U) = \int_0^U f'(s)\eta'(s)ds.$$

Note that η and q satisfy the relation

$$q' = \eta' f'. \tag{3.27}$$

Multiplying both sides of (3.26) by $\eta'(U)$ and using the chain rule and the relation (3.27), we obtain

$$\begin{aligned} & \eta'(U^\varepsilon)U_t^\varepsilon + \eta'(U^\varepsilon)f'(U^\varepsilon)U_x^\varepsilon = \varepsilon\eta'(U^\varepsilon)U_{xx}^\varepsilon \\ \Rightarrow \quad & \eta'(U^\varepsilon)U_t^\varepsilon + q'(U^\varepsilon)U_x^\varepsilon = \varepsilon\eta'(U^\varepsilon)U_{xx}^\varepsilon \\ \Rightarrow \quad & \eta(U^\varepsilon)_t + q(U^\varepsilon)_x = \varepsilon\eta(U^\varepsilon)_{xx} - \varepsilon\eta''(U^\varepsilon)(U_x^\varepsilon)^2. \end{aligned}$$

Since η is a convex function, the second term on the right-hand side is nonpositive. Therefore, we obtain

$$\eta(U^\varepsilon)_t + q(U^\varepsilon)_x \leqslant \varepsilon\eta(U^\varepsilon)_{xx}. \tag{3.28}$$

Therefore, any vanishing viscosity solution $U = \lim_{\varepsilon\to 0} U^\varepsilon$ satisfies

$$\eta(U)_t + q(U)_x \leqslant 0. \tag{3.29}$$

As usual, this expression must be interpreted in the sense of distributions: For all test functions $\varphi \in C_c^1(\mathbb{R} \times [0,\infty))$ with $\varphi \geqslant 0$, U satisfies

$$\int_{\mathbb{R}_+}\int_{\mathbb{R}} \eta(U(x,t))\varphi_t(x,t) + q(U(x,t))\varphi_x(x,t)\,dx\,dt + \int_{\mathbb{R}} \eta(U_0(x))\varphi(x,0)\,dx \geqslant 0. \tag{3.30}$$

The function η is called an *entropy* function and the corresponding function q is called an *entropy flux*. The pair (η, q) is called an *entropy pair*. The inequality (3.29) is referred to as the *entropy condition*, and holds for every entropy pair (η, q). Note here that *every* convex function yields an entropy pair for a scalar conservation law. This is in stark contrast to systems of conservation laws, where there is usually only one entropy pair.

Definition 3.6. *A function $U \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ is an* entropy solution *of* (3.4) *if it satisfies the following conditions:*

(i) U is an weak solution of (3.4)
(ii) U satisfies (3.29) *for all entropy pairs (η, q).*

Any convex function η serves as an entropy function for a scalar conservation law. Of particular importance are the so-called *Kruzkhov entropy pairs*:

$$\eta = \eta(u,c) = |u-c|, \qquad q = q(u,c) = \operatorname{sign}(u-c)(f(u) - f(c)) \tag{3.31}$$

for constants $c \in \mathbb{R}$. The function $\eta(u,c)$ is clearly convex, and it is easy to check that (η, q) is an entropy pair. We say that a function U satisfies the *Kruzkov entropy condition* if it satisfies

$$|U-c|_t + \big(\operatorname{sign}(U-c)(f(U) - f(k))\big)_x \leqslant 0 \tag{3.32}$$

in the weak sense for all constants $c \in \mathbb{R}$, i.e., if (3.30) holds with the pairs $\eta = \eta(U, c)$, $q = q(U, c)$. It is straightforward to see that this guarantees that U is an entropy solution:

Lemma 3.7. *A function $U \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ is an entropy solution of (3.4) if and only if it satisfies the Kruzkov entropy condition.*

Proof. By definition, any entropy solution satisfies the Kruzkov entropy condition, so we only need to prove the opposite implication. Let $a, b \in \mathbb{R}$ be such that $a \leqslant U(x,t) \leqslant b$ for almost every $(x,t) \in \mathbb{R} \times \mathbb{R}_+$. Selecting $c = a$ in (3.32) shows that

$$U_t + f(U)_x \leqslant 0$$

(in the sense of distributions), while selecting $c = b$ gives the opposite inequality. This proves that U is a weak solution.

It is straightforward to check that any convex function $\eta(u)$ can be approximated by a linear combination of functions like $\eta(u, k)$ for $u \in [a, b]$. More precisely, given $\delta > 0$, there exist $N \in \mathbb{N}$, $c_1, \dots, c_N \in \mathbb{R}$, $\alpha_1, \dots, \alpha_N \in \mathbb{R}_+$ and $\beta \in \mathbb{R}$ such that

$$\big|\eta(u) - \eta_\delta(u)\big| \leqslant \delta \qquad \text{for all } u \in [a,b], \text{ where } \eta_\delta(u) := \beta + \sum_{k=1}^{N} \alpha_k |u - c_k|$$

Since the coefficients α_k are positive, the function U satisfies the entropy condition (3.30) also for η_δ, and by passing $\delta \to 0$, we obtain (3.30) for an arbitrary entropy function η. □

Just as with the Rankine–Hugoniot condition, which guarantees that a piecewise C^1 function is a weak solution, it is possible to simplify the entropy condition further:

Theorem 3.8. *Let $\gamma \in C^1(\mathbb{R}_+)$, define $s(t) = \gamma'(t)$ and let $U \in L^\infty(\mathbb{R} \times \mathbb{R}_+)$ be a weak solution of (3.4) which is of the form*

$$U(x,t) = \begin{cases} U^-(x,t) & \text{if } x < \gamma(t) \\ U^+(x,t) & \text{if } x > \gamma(t) \end{cases}$$

where both U^- and U^+ are continuously differentiable functions. Then the following are equivalent:

(i) U is an entropy solution of (3.4), i.e. it satisfies the entropy condition (3.29) for all entropy pairs (η, q)

(ii) At $x = \gamma(t)$, U satisfies

$$[\![q(U)]\!] - s[\![U]\!] \leqslant 0 \tag{3.33}$$

for every entropy pair (η, q)

(iii) For all numbers v between $U^- = U^-(\gamma(t), t)$ and $U^+ = U^+(\gamma(t), t)$,

$$\frac{f(v) - f(U^-)}{v - U^-} \geqslant s \geqslant \frac{f(v) - f(U^+)}{v - U^+} \tag{3.34}$$

(iv) If f is convex or concave, then at $x = \gamma(t)$,

$$f'(U^-) \geqslant s \geqslant f'(U^+). \tag{3.35}$$

Remark 3.9. *The equivalence between (i) and (ii) is due to Eberhard Hopf* [**Hop69**]. *The condition in (iii) is called* Oleinik's condition E *and was used by Olga Oleinik in* [**Ole59**] *to prove uniqueness and L^1 stability of piecewise smooth solutions of scalar, one-dimensional conservation laws. It is an easy exercise to check that* (3.34) *is equivalent to the following: The chord connecting the points $(U^-, f(U^-))$ and $(U^+, f(U^+))$ lies*

- below *the graph of f if $U^- < U^+$*
- above *the graph of f if $U^- > U^+$.*

The condition in (iv) is the Lax entropy condition, *due to Peter Lax* [**Lax57**]. *It has the physical interpretation that "information" (characteristics) can only move* into, *not* out of, *a shock.*

Proof of Theorem 3.8. The equivalence between (i) and (ii) follows, *mutatis mutandis*, the proof of the Rankine–Hugoniot condition (Theorem 3.4), and is left as an exercise to the reader.

For the equivalence of (ii) and (iii) we apply the fundamental theorem of calculus to the quantities $[\![\eta(U)]\!]$ and $[\![q(U)]\!]$ and integrate by parts:

$$\begin{aligned}[\![\eta(U)]\!] &= \int_{U^-}^{U^+} \eta'(v)\,dv = \eta'(v)(v-U^-)\Big|_{v=U^-}^{U^+} - \int_{U^-}^{U^+} \eta''(v)(v-U^-)\,dv \\ &= \eta'(U^+)[\![U]\!] - \int_{U^-}^{U^+} \eta''(v)(v-U^-)\,dv\end{aligned}$$

and

$$\begin{aligned}[\![q(U)]\!] &= \int_{U^-}^{U^+} \eta'(v)f'(v)\,dv = \eta'(v)(f(v)-f(U^-))\Big|_{v=U^-}^{U^+} - \int_{U^-}^{U^+} \eta''(v)(f(v)-f(U^-))\,dv \\ &= \eta'(U^+)[\![f(U)]\!] - \int_{U^-}^{U^+} \eta''(v)(f(v)-f(U^-))\,dv.\end{aligned}$$

Hence,

$$[\![q(U)]\!] - s[\![\eta(U)]\!] = \eta'(U^+)\big([\![f(U)]\!] - s[\![U]\!]\big) + \int_{U^-}^{U^+} \eta''(v)\big(s(v-U^-) - (f(v)-f(U^-))\big)\,dv.$$

The first term vanishes because of the Rankine–Hugoniot condition (3.16). If the remaining integral is to be nonpositive for all convex entropies η, then we must have

$$\operatorname{sign}\big(U^+-U^-\big)\big(s(v-U^-) - (f(v)-f(U^-))\big) \leqslant 0$$

for all v between U^- and U^+, which using the Rankine–Hugoniot condition is precisely (3.34).

Finally, for the equivalence between (iii) and (iv) we assume that f is convex; the concave case follows similarly. Then the left- and right-hand sides of (3.34) are monotone functions of v, so it suffices to check (3.34) in the limits $v \to U^-$ and $v \to U^+$, respectively. But taking these limits reduces (3.34) to (3.35), so we are done. □

3.3.2. Stability estimates. The entropy inequality (3.29) can be used to obtain stability estimates on solutions. To see this, we do the following *formal* computation: Integrate (3.29) in space and integrate by parts to obtain

$$\frac{d}{dt}\int_{\mathbb{R}} \eta(U)\,dx \leqslant 0 \quad \Rightarrow \quad \int_{\mathbb{R}} \eta(U(x,t))\,dx \leqslant \int_{\mathbb{R}} \eta(U_0(x))\,dx. \tag{3.36}$$

Since the function η may be any convex function, we can choose $\eta(U) = U^2$ and obtain a bound on entropy solutions in L^2. This estimate is a *nonlinear* analogue of the energy estimate (2.6) for the linear transport equation. Choosing η as $\eta(U) = |U|^p$ will lead to the L^p estimate $\|U(t)\|_{L^p(\mathbb{R})} \leqslant \|U_0\|_{L^p(\mathbb{R})}$ for any $p \in [1,\infty)$. Taking the limit $p \to \infty$ yields the uniform bound $\|U(t)\|_{L^\infty(\mathbb{R})} \leqslant \|U_0\|_{L^\infty(\mathbb{R})}$.

The above L^p estimates give bounds on the *amplitude* of U, but we can also use the entropy condition to derive bounds on the *derivative*. This will give a restriction on the amount of oscillations in U, and will be important for the stability and convergence analysis of numerical schemes. Let g be a function defined on an interval $[a,b]$. The *total variation* of g is defined as

$$\|g\|_{TV([a,b])} = \sup_{\mathcal{P}} \sum_{j=1}^{N-1} |g(x_{j+1}) - g(x_j)|, \tag{3.37}$$

where the supremum is taken over all partitions $\mathcal{P} = \{a = x_1 < x_2 < \cdots < x_N = b\}$ of the interval $[a,b]$. It is straightforward to check that if g is differentiable, then

$$\|g\|_{TV([a,b])} = \int_a^b \left|\frac{dg}{dx}\right| dx.$$

The total variation is only a semi-norm, because the total variation of any constant function is zero. We turn it into a norm by defining

$$\|g\|_{BV([a,b])} = \|g\|_{L^1([a,b])} + \|g\|_{TV([a,b])}. \tag{3.38}$$

We define the space of functions with bounded variation (BV) on $\mathbb{R}$ as

$$BV(\mathbb{R}) = \left\{ g \in L^1(\mathbb{R}) \;:\; \|g\|_{BV(\mathbb{R})} < \infty \right\}. \tag{3.39}$$

We are now in a position to state the main well-posedness results for scalar conservation laws:

Theorem 3.10. *Assume that $f \in C^1(\mathbb{R})$ and $U_0 \in L^1(\mathbb{R}) \cap L^\infty(\mathbb{R})$. Then there exists a unique entropy solution U of (3.4), and U satisfies the following:*

- L^1 bound:

$$\|U(\cdot,t)\|_{L^1} \leqslant \|U_0\|_{L^1} \tag{3.40}$$

- L^∞ bound:

$$\|U(\cdot,t)\|_{L^\infty} \leqslant \|U_0\|_{L^\infty}, \tag{3.41}$$

- TV bound: *If $\|U_0\|_{TV} < \infty$ then*

$$\|U(\cdot,t)\|_{TV} \leqslant \|U_0\|_{TV}. \tag{3.42}$$

- Time continuity: *If $\|U_0\|_{TV} < \infty$ then*

$$\|U(t) - U(s)\|_{L^1(\mathbb{R})} \leqslant |t-s| M \|U_0\|_{TV(\mathbb{R})} \tag{3.43}$$

where

$$M = M(U_0) := \max_{\underline{u} \leqslant u \leqslant \overline{u}} |f'(u)|, \quad \underline{u} := \operatorname*{ess\,inf}_{x\in\mathbb{R}} U_0(x),\ \overline{u} := \operatorname*{ess\,sup}_{x\in\mathbb{R}} U_0(x).$$

Furthermore, if U and V are the entropy solutions of (3.4) with initial U_0 and V_0, respectively, then the following hold:

- L^1 stability:

$$\|U(\cdot,t) - V(\cdot,t)\|_{L^1(\mathbb{R})} \leqslant \|U_0 - V_0\|_{L^1(\mathbb{R})} \qquad \text{for all } t > 0. \tag{3.44}$$

- Local L^1 stability: *For any $a < b$,*

$$\int_a^b |U(x,t) - V(x,t)|\,dx \leqslant \int_{a-Mt}^{b+Mt} |U_0(x) - V_0(x)|\,dx \qquad \text{for all } t > 0 \tag{3.45}$$

where $M = \max(M(U_0), M(V_0))$.

- Monotonicity: *If $U_0(x) \leqslant V_0(x)$ for all $x \in \mathbb{R}$, then*

$$U(x,t) \leqslant V(x,t) \qquad \forall\ x \in \mathbb{R}. \tag{3.46}$$

Sketch of proof. The existence of entropy solutions follow from the vanishing viscosity approximation: Let U^ε be solutions of (3.26) for $\varepsilon > 0$. From the computations in Section 3.3.1, the vanishing viscosity solution $U = \lim_{\varepsilon\to 0} U^\varepsilon$ satisfies the entropy condition (3.29). By the maximum principle for (3.26), we have $\|U^\varepsilon(t)\|_{L^\infty(\mathbb{R})} \leqslant \|U_0\|_{L^\infty(\mathbb{R})}$ for every $t > 0$ and $\varepsilon > 0$. Taking the limit $\varepsilon \to 0$, we conclude that the vanishing viscosity solution U is an entropy solution which satisfies the L^∞ bound (3.41).

Let U and V be entropy solutions of (3.4) with initial data U_0 and V_0. By Lemma 3.7, this is equivalent to stating that the following inequalities hold (in the sense of distributions):

$$\begin{aligned} \partial_t \eta(U,c) + \partial_x q(U,c) &\leqslant 0 \qquad \forall\ c \in \mathbb{R} \\ \partial_t \eta(d,V) + \partial_x q(d,V) &\leqslant 0 \qquad \forall\ d \in \mathbb{R} \end{aligned} \tag{3.47}$$

(here we have used the fact that η and q are symmetric in both variables; cf. (3.31)). We now compute the time derivative of $\eta(U,V) = |U-V|$:

$$\begin{aligned} \partial_t \eta(U,V) &= \partial_t \eta(U,c)\big|_{c=V} + \partial_t \eta(d,V)\big|_{d=U} \\ &\leqslant -\partial_x q(U,c)\big|_{c=V} - \partial_x q(d,V)\big|_{d=U} \\ &= -\partial_x q(U,V), \end{aligned}$$

where we have first used the chain rule, then (3.47) and then the chain rule again. Thus,

$$\partial_t |U-V| + \partial_x q(U,V) \leqslant 0 \tag{3.48}$$

holds in the sense of distributions. Now integrate this inequality over the trapezoid

$$\big\{(x,t) : 0 \leqslant t \leqslant T,\ a - M(T-t) \leqslant x \leqslant b + M(T-t)\big\}$$

for some $a < b$ and $T > 0$. After applying the divergence theorem we get the inequality

$$\int_a^b |U-V|(x,t)\,dx - \int_{a-Mt}^{b+MT} |U_0 - V_0|(x))\,dx \\ - \int_0^T \big(q(U,V) - M\eta(U,V)\big)(x_L(t),t) + \big(q(U,V) - M\eta(U,V)\big)(x_R(t),t)\,dt \leqslant 0 \tag{3.49}$$

where $x_L(t) = a - M(T-t)$ and $x_R(t) = b + M(T-t)$. From the definition (3.31) of η and q, it is straightforward to see that

$$q(U,V) - M\eta(U,V) = |U-V|\left(\frac{q(U,V)}{|U-V|} - M\right) \leqslant 0.$$

Applying this to (3.49) we conclude that (3.45) holds. Passing $a \to -\infty$, $b \to \infty$ yields (3.44).

The L^1 stability property (3.44) now implies the remaining properties. If we set $U_0 = V_0$ we find that $U(\cdot,t) = V(\cdot,t)$ for all times $t > 0$, and hence the entropy solution is unique (and so must correspond to the vanishing viscosity solution). If $V_0 \equiv 0$ then the corresponding entropy solution is $V \equiv 0$, which yields (3.40). If $V_0(x) = U_0(x+h)$ then $V(x,t) = U(x+h,t)$ is the corresponding entropy solution, so that (3.44) implies

$$\int_{\mathbb{R}} |U(x+h,t) - U(x,t)|\,dx \leqslant \int_{\mathbb{R}} |U_0(x+h) - U_0(x)|\,dx.$$

Dividing by h and passing $h \to 0$ yields (3.42).

To prove the monotonicity property (3.46), we observe that (3.44), together with the facts that $\int_{\mathbb{R}} U(x,t)\,dx = \int_{\mathbb{R}} U_0(x)\,dx$ and $\int_{\mathbb{R}} V(x,t)\,dx = \int_{\mathbb{R}} V_0(x)\,dx$, imply that

$$\int_{\mathbb{R}} \big(U(x,t) - V(x,t)\big)^+\,dx \leqslant \int_{\mathbb{R}} \big(U_0(x) - V_0(x)\big)^+\,dx,$$

where $a^+ = \max(a,0) = \frac{|a|+a}{2}$. If $U_0 \leqslant V_0$ then $\big(U_0(x) - V_0(x)\big)^+ \equiv 0$ for all $x \in \mathbb{R}$, so $\int_{\mathbb{R}} \big(U(x,t) - V(x,t)\big)^+\,dx = 0$, whence (3.46) follows.

We will not prove the time continuity property (3.43) here. However, we will see in Section 4.5 that it is straightforward to prove a discrete version of time continuity for monotone finite volume schemes. The convergence of these schemes to the entropy solution then implies the same property for the entropy solution. □

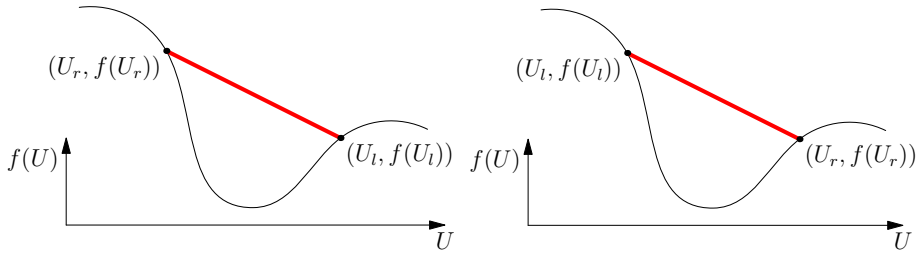
Remark 3.11. *The above is admittedly only a formal proof, but it contains all of the essential features of the full, rigorous proof. For a rigorous study of the viscous regularization* (3.26) *and its $\varepsilon \to 0$ limit, consult e.g.* [**HR15**, Appendix B] *or* [**GR91**, Section II.2]. *The rigorous proof of the essential estimate* (3.48) *is based on the ingenious* doubling of variables *idea of Kruzkhov* [**Kru70**]*, which is well-worth a closer study for anyone interested in PDE techniques.*

The stability bound (3.45) *can be interpreted as follows: Information propagates at a finite speed, which can be bounded by $M = \max_u |f'(u)|$. This is in contrast with e.g. parabolic PDEs, which have an infinite speed of propagation.*

3.4. Solutions to the Riemann problem for general f

Armed with Oleinik's condition E (3.34) we can now tackle the Riemann problem when $f \in C^1(\mathbb{R})$ is a general, not necessarily convex, function. Recall (see Remark 3.9) that Oleinik's condition E can equivalently be formulated as follows: The chord joining $(U_L, f(U_L))$ and $(U_R, f(U_R))$ must lie below the graph of the function f between these points when $U_L < U_R$, and should lie above the graph if $U_L > U_R$.

We have the following recipe for constructing a weak solution for the Riemann problem (3.13) for the conservation law (3.4), that satisfies Oleinik's condition E. Without loss of generality, we may assume that $U_L < U_R$. In order to satisfy Oleinik's condition E, we have to consider the *lower convex envelope* f_c of f between U_L and U_R. The lower convex envelope of a function f is the largest convex function (largest in the pointwise sense) that is everywhere smaller than or equal to f (see Figure 3.10). Analogously, the upper concave envelope of f is the smallest concave function that is larger than or equal to f.



(a) Admissible (b) Inadmissible

FIGURE 3.9. Admissible and inadmissible shocks under Oleinik's condition E.

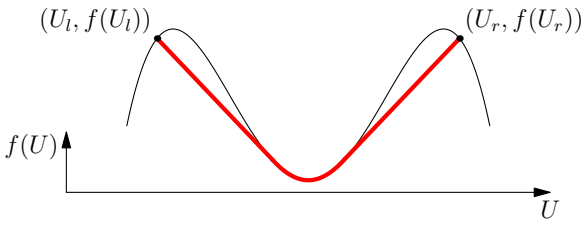
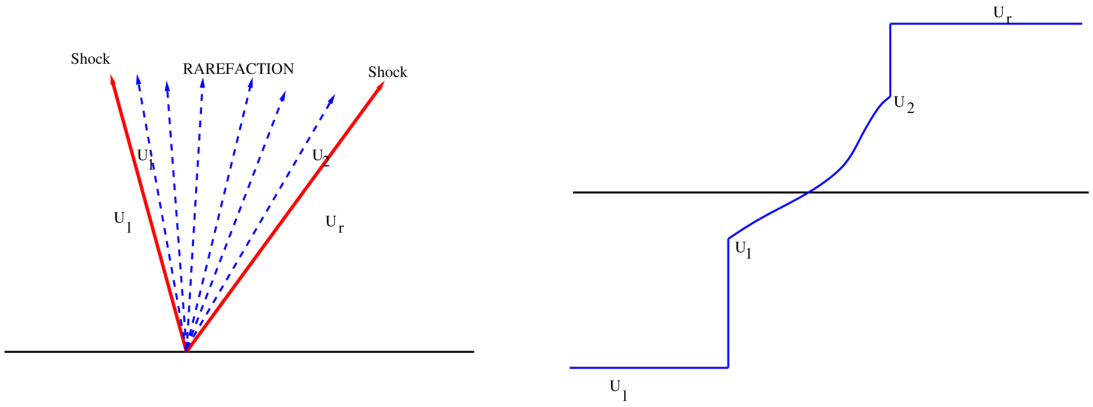


FIGURE 3.10. The solution of the Riemann problem with a non-convex flux. The lower convex envelope is the thick red curve. Solutions are constructed as shocks, followed by rarefactions.

The domain $[U_L, U_R]$ is divided into two sets of regions, one in which $f_c = f$ and another with $f_c \neq f$. In the second region, f_c is affine. The strategy for constructing an entropy solution is to join U_L and U_R by rarefaction waves and shocks. Shocks are used in the affine region and rarefactions in the complement. The solution of the Riemann problem is then (3.25) with f replaced by f_c. An illustration is provided in Figure 3.10.



(a) Solutions in x-t-plane (b) $U(\cdot, t)$

FIGURE 3.11. Entropy solutions for the Riemann problem with a non-convex flux. Left: Solutions in space-time. Right: A snapshot of the solution illustrating compound shocks.

When $U_L > U_R$, the upper concave envelope can be analogously used. Details of this construction can be obtained from [**GR91**, Section II.6]. A wave consisting of rarefaction, followed immediately by a shock (or vice versa) is termed a *compound shock* (see Figure 3.11).

3.5. Summary

Summarizing the theoretical discussion of this section, we have the following results:

- Solutions of the conservation law (3.4) may develop discontinuities or shock waves, even for smooth initial data. Consequently, weak solutions are sought. Shock speeds are computed with the Rankine–Hugoniot condition (3.16).
- Weak solutions are not necessarily unique. Entropy conditions like Oleinik's condition E have to be imposed. Self-similar continuous solutions or rarefaction waves have to be considered.
- Explicit solutions for the Riemann problem (even for non-convex fluxes) can be constructed in terms of shocks, rarefaction waves and compound shocks.
- Entropy solutions exist, are unique and are stable in L^1 with respect to the initial data. Furthermore, the entropy solutions satisfy an L^∞ estimate, L^p estimates and are Total Variation Diminishing (TVD)—that is, the total variation decreases in time.

CHAPTER 4

Finite volume schemes for scalar conservation laws

In this chapter we will design efficient schemes for the scalar conservation law

$$U_t + f(U)_x = 0. \tag{4.1}$$

The discussion on the linear transport equation

$$U_t + aU_x = 0 \tag{4.2}$$

shows that central differences cannot be used to approximate the conservation law, even in the simplest case of linear transport. For linear transport equations, the crucial step in designing an efficient scheme was to *upwind* it by taking derivatives in the direction of information propagation. For a linear equation with constant coefficients like (4.2), the direction of information propagation is given by the constant velocity field. For a nonlinear conservation law like (4.1), the wave speeds depend on the solution itself and can not be determined a priori. Thus, it is not clear how differences can be upwinded.

Another issue is the very nature of finite difference approximations like (2.16). The key idea underlying finite difference schemes is to replace the derivatives in equations like (4.1) with a finite difference. This procedure requires the solutions to be smooth and the equation to be satisfied point-wise. However, the solutions to the scalar conservation law (4.1) are not necessarily smooth and so the Taylor expansion – essential for replacing derivatives with finite differences – is no longer valid. Hence, the finite difference framework is not suited for approximating conservation laws. Instead, we need to develop a new paradigm for designing numerical schemes for scalar conservation laws.

4.1. Finite volume scheme

The first step in any numerical approximation is to discretize the computational domain in both space and time.

4.1.1. The grid. For simplicity, we consider a uniform discretization of the domain $[x_L, x_R]$. The discrete points are denoted as $x_j = x_L + (j + 1/2)\Delta x$ for $j = 0, \ldots, N$, where $\Delta x = \frac{x_R - x_L}{N+1}$. We also define the midpoint values

$$x_{j-1/2} = x_j - \Delta x/2 = x_L + j\Delta x$$

for $j = 0, \ldots, N+1$. These values define computational cells or *control volumes*

$$\mathcal{C}_j = \left[x_{j-1/2},\ x_{j+1/2}\right).$$

As we will see soon, the finite volume method uses the control volumes $\mathcal{C}_j$ instead of the mesh points x_j. We use a uniform discretization in time with time step Δt. The time levels are denoted by $t^n = n\Delta t$. See Figure 4.1 for an illustration of the grid.

4.1.2. Cell averages. A finite difference method is based on approximating the point values of the solution of a PDE. This approach is not suitable for conservation laws as the solutions are not continuous and point values may not make sense. Instead, we change the perspective and use the *cell averages*

$$U_j^n \approx \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} U(x, t^n)\, dx \tag{4.3}$$

at each time level t^n as the main object of interest for our approximation.

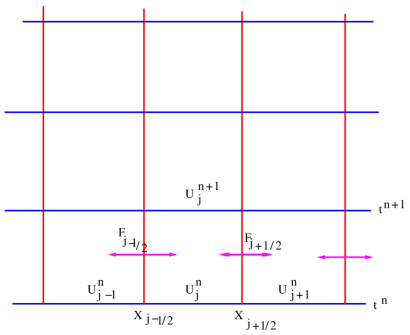


FIGURE 4.1. A typical finite volume grid displaying cell averages and fluxes.

The cell average (4.3) is well defined for any integrable function, hence also for the solutions of the conservation law (4.1). The aim of the finite volume method is to update the cell average of the unknown at every time step, starting with

$$U_j^0 = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} U_0(x)\, dx. \tag{4.4}$$

4.1.3. Integral form of the conservation law. Assume that the cell averages U_j^n at some time level t^n are known. How do we obtain the cell averages U_j^{n+1} at the next time level t^{n+1}? A finite volume method computes the cell average at the next time level by integrating the conservation law (4.1) over the domain $[x_{j-1/2},\ x_{j+1/2}) \times [t^n,\ t^{n+1})$. This gives

$$\int_{t^n}^{t^{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} U_t\, dxdt + \int_{t^n}^{t^{n+1}} \int_{x_{j-1/2}}^{x_{j+1/2}} f(U)_x\, dxdt = 0.$$

Using the fundamental theorem of calculus gives

$$\begin{aligned} &\int_{x_{j-1/2}}^{x_{j+1/2}} U\left(x, t^{n+1}\right)\, dx - \int_{x_{j-1/2}}^{x_{j+1/2}} U\left(x, t^n\right) dx \\ &= -\int_{t^n}^{t^{n+1}} f\left(U(x_{j+1/2}, t)\right)\, dt + \int_{t^n}^{t^{n+1}} f\left(U(x_{j-1/2}, t)\right)\, dt. \end{aligned} \tag{4.5}$$

Defining

$$\bar{F}_{j+1/2}^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\big(U(x_{j+1/2}, t)\, dt \tag{4.6}$$

and dividing both sides of (4.5) by Δx, we obtain

$$U_j^{n+1} = U_j^n - \frac{\Delta t}{\Delta x} \left(\bar{F}_{j+1/2}^n - \bar{F}_{j-1/2}^n\right). \tag{4.7}$$

Equation (4.7) is a statement of conservation: The change of the cell average is given by the difference in fluxes across the boundary of the cell. See Figure 4.1 for an illustration. Note that the relation (4.7) is not explicit as $\bar{F}$ need a priori knowledge of the exact solution. The main ingredient in a finite volume scheme is a clever procedure to approximate these fluxes.

4.1.4. Godunov method. Godunov [**God59**] came up with an ingenious idea for approximating the numerical fluxes in (4.7). We wish to approximate

$$\bar{F}_{j+1/2}^n = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} f\big(U(x_{j+1/2}, t)\big)\, dt$$

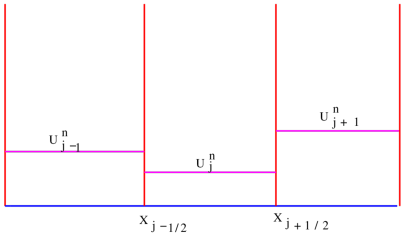


FIGURE 4.2. Cell averages define Riemann problems at every interface.

at each interface $x_{j+1/2}$. As the cell averages U_j^n are constant in each cell $\mathcal{C}_j$ at each time level, Godunov observed that they define at each cell interface $x_{i+1/2}$ a *Riemann problem*

$$\begin{cases} U_t + f(U)_x = 0 \\ U(x,t^n) = \begin{cases} U_j^n & \text{if } x < x_{j+1/2} \\ U_{j+1}^n & \text{if } x > x_{j+1/2}. \end{cases} \end{cases} \tag{4.8}$$

Thus at every time level, the cell averages define a superposition of Riemann problems of the form (4.8) at each interface (see Figure 4.2). In the previous chapter, we have solved Riemann problems like (4.8) explicitly. The solution consists of shock waves, rarefactions and compound waves. Hence, the Riemann problem at every time level can be solved explicitly in terms of waves, emanating from each interface (Figure 4.3). Furthermore, the solution of each Riemann problem in (4.8) is self-similar, that is, the solution $\bar{U}_j(x,t)$ of (4.8) can be written as a function $\bar{U}(\xi)$ of a single variable $\xi = \frac{x-x_{j+1/2}}{t-t^n}$,

$$\bar{U}_j(x,t) = \bar{U}_j\left(\frac{x-x_{j+1/2}}{t-t^n}\right). \tag{4.9}$$

Waves from neighboring Riemann problems can intersect after some time (Figure 4.3(a)). However, each wave has a finite speed of propagation and the maximum wave speed of any Riemann problem is bounded by

$$\max_j |f'(U_j^n)|$$

(see Chapter 3). Hence, imposing the *CFL condition*

$$\max_j |f'(U_j^n)| \frac{\Delta t}{\Delta x} \leqslant \frac{1}{2} \tag{4.10}$$

ensures that waves from neighboring problems do not interact before reaching the next time level (see Figure 4.3(b))[1] . Assume now that this condition is satisfied. By (4.9), the solution is constant when ξ is constant, so in particular, at the cell interface $\xi = 0$, the flux across the interface is given by the constant value

$$f\big(U(x_{j+1/2},t)\big) = f\big(\bar{U}_j(0)\big).$$

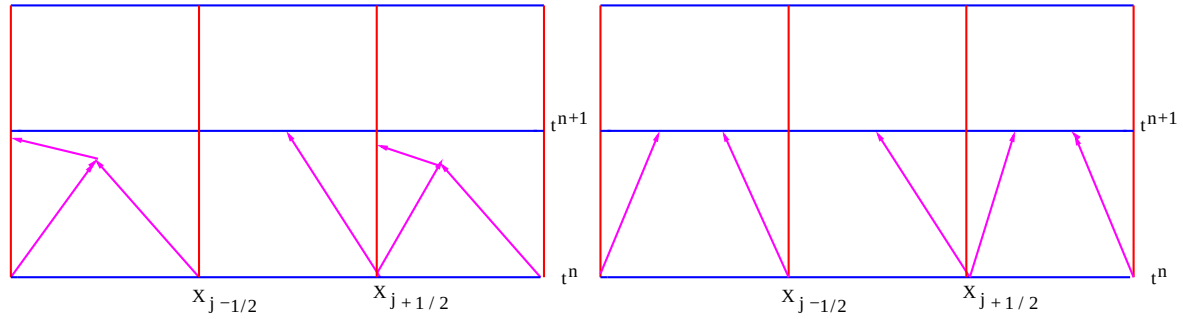
At $\xi = 0$ (corresponding to the curve $x = x_{j+1/2}\ \forall\ t > t^n$), the function $\bar{U}_j(\xi)$ is either continuous or discontinuous. If $\bar{U}_j$ is continuous at $\xi = 0$, we obviously have

$$f\big(\bar{U}_j(0+)\big) = f\left(\bar{U}_j(0-)\right). \tag{4.11}$$

On the other hand, if $\bar{U}_j$ is discontinuous at $\xi = 0$, then we have a discontinuity along the line $x = x_{j+1/2}$ for all $t > t^n$, in other words, a stationary shock located at the cell interface. Since the discontinuity must satisfy the Rankine–Hugoniot condition (3.16), we have

$$f\big(\bar{U}_j(0+)\big) - f\left(\bar{U}_j(0-)\right) = 0 \cdot \big(\bar{U}_j(0+) - \bar{U}_j(0-)\big) = 0,$$

[1] In fact, we can use the less strict requirement $\max_j |f'(U_j^n)| \frac{\Delta t}{\Delta x} \leqslant 1$, but to keep some arguments simple, we stick to the strict version.



(a) Interacting Riemann waves. (b) Non-interacting waves.

FIGURE 4.3. Left: Waves of Riemann problems from neighboring interface can interact after some time. Right: The waves can be prevented from interacting before time Δt by the CFL condition (4.10)

and so (4.11) holds also in this case. Hence, the term $f(\bar{U}_j(0))$ is well-defined, and we may define the edge-centered flux value

$$F^n_{j+1/2} := f\big(\bar{U}_j(0+)\big) = f\left(\bar{U}_j(0-)\right). \tag{4.12}$$

In conclusion, the approximate flux in (4.6) is constant in time and can be explicitly computed as

$$\bar{F}^n_{j+1/2} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f\big(U(x_{j+1/2},t)\big)\,dt = F^n_{j+1/2}, \tag{4.13}$$

with $F^n_{j+1/2}$ being the Riemann flux (4.12). Substituting (4.13) in (4.7) leads to the finite volume scheme

$$U^{n+1}_j = U^n_j - \frac{\Delta t}{\Delta x}\left(F^n_{j+1/2} - F^n_{j-1/2}\right). \tag{4.14}$$

The form (4.14) is the standard form of a finite volume scheme for conservation laws. The numerical flux F is given in terms of the Riemann solution and can be explicitly computed for scalar conservation laws.

4.1.5. Godunov flux. It turns out that we can compute explicit formulas for the numerical flux in (4.14). To this end, we need to obtain the value of the flux of the Riemann problem (4.8) at the interface $x_{j+1/2}$. A lengthy computation based on a case by case analysis leads to the formula

$$F^n_{j+1/2} = F\big(U^n_j, U^n_{j+1}\big) = \begin{cases} \min\limits_{U^n_j \leqslant \theta \leqslant U^n_{j+1}} f(\theta) & \text{if } U^n_j \leqslant U^n_{j+1} \\ \max\limits_{U^n_{j+1} \leqslant \theta \leqslant U^n_j} f(\theta) & \text{if } U^n_j > U^n_{j+1}. \end{cases} \tag{4.15}$$

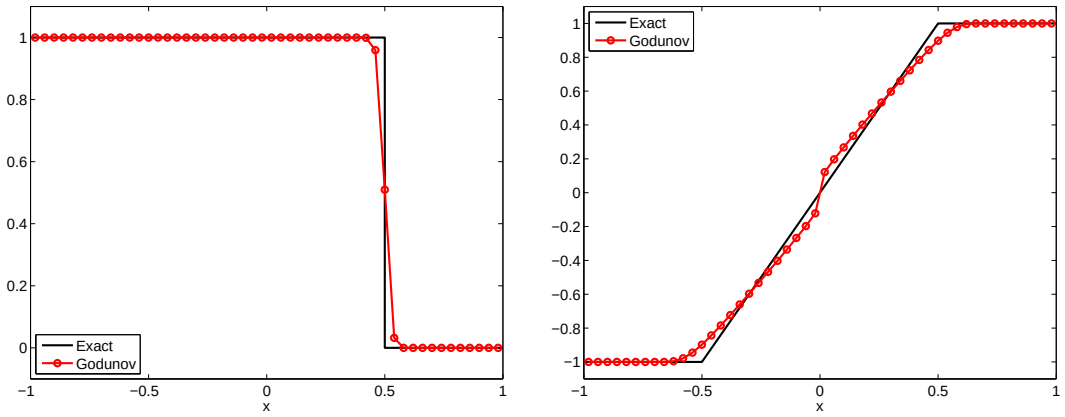
This formula is valid also for non-convex flux functions. The *Godunov scheme* is (4.14) with the Godunov flux (4.15).

Exercise 4.1. *Computing the flux* (4.15) *can be complicated, since an optimization problem has to be solved. Show that in the special case where the flux function f has a single minimum at the point ω and no local maxima, the formula* (4.15) *can be simplified to*

$$F^n_{j+1/2} = F(U^n_j, U^n_{j+1}) = \max\Big(f\left(\max\left(U^n_j,\ \omega\right)\right),\ f\left(\min\left(U^n_{j+1},\ \omega\right)\right)\Big). \tag{4.16}$$

Note that strictly convex functions have this property. The formulas for the case of a flux with a single maximum and no minima are obtained analogously.

Exercise 4.2. *Show that for the linear transport equation* (4.2)*, the Godunov scheme* (4.14)*,* (4.15) *is identical to the standard upwind scheme* (2.16)*. Thus, the Godunov scheme can be viewed as a generalization of the upwind scheme to nonlinear scalar conservation laws.*



(a) Initial data (4.17) at time $t = 1$.

(b) Initial data (4.18) at time $t = 0.5$.

FIGURE 4.4. Approximate solution for Burgers equation with the Godunov scheme with 50 mesh points. `[burgers_disc.m]`

4.1.6. Numerical experiments. Consider Burgers' equation (3.3) with Riemann data

$$U(x,0) = \begin{cases} 1 & \text{if } x < 0 \\ 0 & \text{if } x > 0. \end{cases} \tag{4.17}$$

In this case, the exact solution is given by a single shock connecting 1 and 0, traveling at speed of $1/2$ (see Chapter 3). Numerical solutions with the Godunov scheme (4.14), (4.15) with 50 mesh points are plotted in Figure 4.4 (a). The results show that the solution is approximated very well, with the shock being resolved sharply. The numerical solutions do not oscillate or show any anomalies or instabilities.

Next, we test Burgers' equation with initial data

$$U(x,0) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x > 0. \end{cases} \tag{4.18}$$

The exact solution in this case is given by a rarefaction wave (3.25). The approximate solutions using the Godunov scheme are plotted in Figure 4.4 (b). Again the results demonstrate that the Godunov scheme is stable and robust.

As a final test case, we consider Burgers' equation with initial data

$$U(x,0) = \sin(4\pi x) \qquad \text{for } -1 \leqslant x \leqslant 1. \tag{4.19}$$

The initial data is a sine wave and it is much more difficult to write down an explicit formula for the solution. Instead, we compute this configuration with the Godunov scheme using periodic boundary conditions. The results are shown in Figure 4.5. A reference solution computed on a very fine mesh (5000 points) is also shown for the sake of comparison. The behavior of the solution is quite complicated. The initial sine wave compresses in some parts and expands in some other parts, leading to a combination of shocks and rarefactions. The solution finally decays into a so-called *N-wave*. The Godunov scheme provides a good approximation to this complicated solution.

4.1.7. Beyond the Godunov Scheme. The Godunov scheme (4.14), (4.15) has many desirable properties as demonstrated by numerical experiments. However, it does present a few problems:

- It relies on the availability of an *explicit* formula for the solutions of the Riemann problem. In the case of scalar conservation law (4.1), we are lucky to have such formulas at hand. However, more complicated systems of conservation laws may not yield such formulas.
- The only information needed in the numerical flux (4.12) is the value of the flux at the interface. Solving the entire Riemann problem for the sake of this value seems unnecessary.

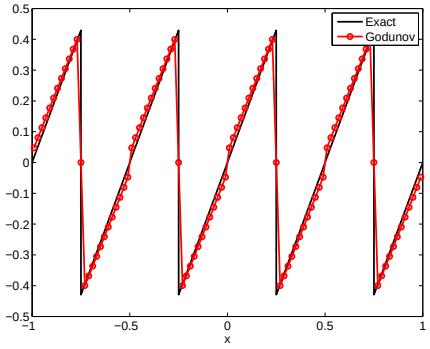


FIGURE 4.5. Approximate solution for Burgers' equation with the Godunov scheme at time $t = 0.5$ with 50 mesh points with initial data (4.19). `[burgers_godunov_sine.m]`

- At the level of implementation, the formula (4.16) provides a simple characterization of the Godunov flux for a large class of flux functions. However, more complicated flux functions with a large number of extremal points need the solution of an optimization problem. Such a problem might be very computationally costly.

These factors encourage the search for alternative numerical fluxes in (4.14).

4.2. Approximate Riemann Solvers

Since we are interested in approximating the solutions of the conservation law (4.1), it seems reasonable to replace the exact solutions of the Riemann problem (4.8) (used in the Godunov scheme) with approximate solutions. These approximate solutions can then be used to define the numerical flux F as in (4.13). Such schemes which replace the exact solutions of the Riemann problem (4.8) with approximations called *approximate Riemann solvers*. We present some of them below.

4.2.1. Linearized (Roe) solvers. Our aim is to approximate the solutions of the Riemann problem (4.8). A common method for solving nonlinear equations is to *linearize* them. Linearization entails replacing the nonlinear flux function in (4.1) with a locally linearized version,

$$f(U)_x = f'(U)U_x \approx \hat{A}_{j+1/2}U_x, \tag{4.20}$$

where $\hat{A} \approx f'$ is a constant state around which the nonlinear flux function is linearized. There are many possible candidates for the linearizing state, one simple choice being

$$\hat{A}_{j+1/2} = f'\left(\frac{U_j^n + U_{j+1}^n}{2}\right),$$

the flux of the arithmetic average of the two constant states. We will use a more sophisticated *Roe average*:

$$\hat{A}_{j+1/2} = \begin{cases} \frac{f(U_{j+1}^n) - f(U_j^n)}{U_{j+1}^n - U_j^n} & \text{if } U_{j+1}^n \neq U_j^n \\ f'(U_j^n) & \text{if } U_{j+1}^n = U_j^n. \end{cases} \tag{4.21}$$

Note that the Roe average also represents a linear approximation of f'. The numerical flux F is obtained by replacing the Riemann problem (4.8) with a linearized Riemann problem,

$$\begin{cases} U_t + \hat{A}_{j+1/2}U_x = 0 \\ U(x, t^n) = \begin{cases} U_j^n & \text{if } x < x_{j+1/2} \\ U_{j+1}^n & \text{if } x > x_{j+1/2}. \end{cases} \end{cases} \tag{4.22}$$

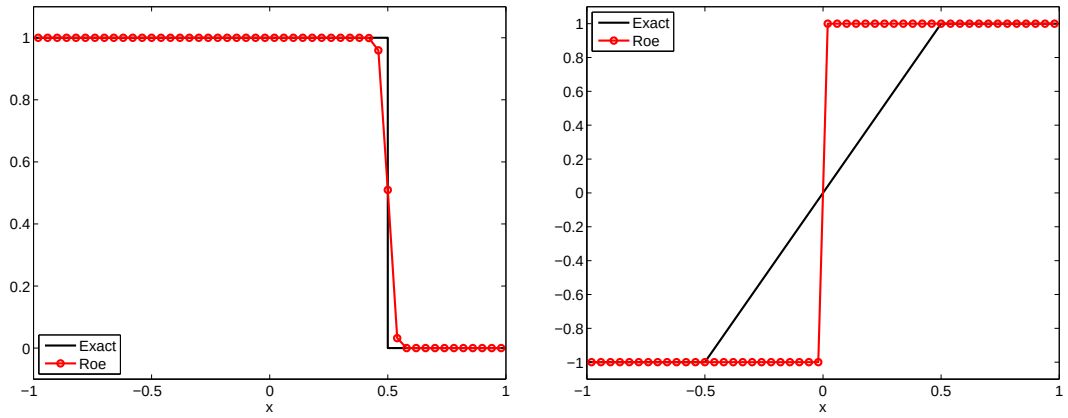
This Riemann problem is very simple to solve as it involves a linear transport equation with a constant velocity field. Solving it explicitly we obtain the formula

$$F_{j+1/2}^n = F^{\text{Roe}}(U_j^n, U_{j+1}^n) = \begin{cases} f(U_j^n) & \text{if } \hat{A}_{j+1/2} \geqslant 0 \\ f(U_{j+1}^n) & \text{if } \hat{A}_{j+1/2} < 0. \end{cases} \tag{4.23}$$

The finite volume scheme (4.14) with the Roe flux (4.23) is termed the *Roe* or *Murman-Roe* scheme. It is simpler to implement when compared to the Godunov scheme as no optimization problem needs to be solved.

Numerical results with the Roe scheme for Burgers' equation with Riemann data (4.17) are shown in Figure 4.6(a). They show that the Roe scheme approximates the shock as accurately as the Godunov scheme.

Figure 4.6 (b) shows numerical results for the Riemann data (4.18). In this case, the Roe scheme fails completely and approximates the wrong stationary shock solution. The same stationary solution persists even when the mesh is refined. Thus, the Roe scheme leads to numerical artifacts for some problems. This failure will be analyzed in detail in the sequel.



(a) Shock solution with initial data (4.17) at $t = 1$. (b) Rarefaction wave solution initial data (4.18) at $t = 0.5$.

FIGURE 4.6. Approximate solutions for Burgers equation with the Roe scheme with 50 mesh points. `[burgers_disc.m]`

4.2.2. Central schemes. The Roe scheme fails at resolving rarefactions. Due to linearization, the solution of the approximate Riemann problem (4.22) only consists of a single wave that travels to the right or to the left, depending on the sign of the Roe average $\hat{A}$. When the exact solutions of Riemann problems for the conservation law consists of shocks, then the solution is a single, either left- or right-going, wave. However, the situation with rarefactions is very different. The rarefaction wave that solves (4.18) can travel in both directions (see Figure 4.7). As we have seen, the Roe scheme may be unable to capture such behavior.

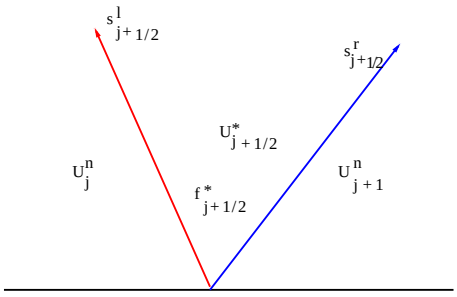


FIGURE 4.7. An approximate Riemann solver with bi-directional waves.

Instead of linearizing the conservation law, we approximate the solutions of the Riemann problem by replacing the exact solution with *two waves*, one traveling to the left of the interface with speed $s^l_{j+1/2}$ and another to the right with speed $s^r_{j+1/2}$ (see Figure 4.7). The speeds will be specified later on.

We approximate the solution of (4.8) with

$$U(x,t) = \begin{cases} U_j^n & \text{if} \quad x < s_{j+1/2}^l t \\ U_{j+1/2}^* & \text{if} \quad s_{j+1/2}^l t < x < s_{j+1/2}^r t \\ U_{j+1}^n & \text{if} \quad x > s_{j+1/2}^r t. \end{cases} \tag{4.24}$$

Thus, the exact solution is replaced by two waves separated by a middle state. The middle state can be determined by local conservation using the Rankine–Hugoniot conditions (3.16):

$$\begin{aligned} f(U_{j+1}^n) - f_{j+1/2}^* &= s_{j+1/2}^r \big(U_{j+1}^n - U_{j+1/2}^*\big), \\ f(U_j^n) - f_{j+1/2}^* &= s_{j+1/2}^l \big(U_j^n - U_{j+1/2}^*\big), \end{aligned} \tag{4.25}$$

where $f_{j+1/2}^*$ is the intermediate flux (see Figure 4.7). Observe that we require f^* to be an independent variable. Thus, (4.25) represents a system of two linear equations for two unknowns that can be solved exactly to obtain

$$f_{j+1/2}^* = \frac{s_{j+1/2}^r f\big(U_j^n\big) - s_{j+1/2}^l f\big(U_{j+1}^n\big) + s_{j+1/2}^r s_{j+1/2}^l \big(U_{j+1}^n - U_j^n\big)}{s_{j+1/2}^r - s_{j+1/2}^l}. \tag{4.26}$$

In particular, if we choose the speeds to be equal but of opposite sign, so $s^r = -s^l = s$, then (4.26) reduces to

$$f_{j+1/2}^* = \frac{f(U_j^n) + f(U_{j+1}^n)}{2} - \frac{s_{j+1/2}}{2}\left(U_{j+1}^n - U_j^n\right). \tag{4.27}$$

In either case, the numerical flux is given by

$$F_{j+1/2}^n = F\left(U_j^n, U_{j+1}^n\right) = f_{j+1/2}^*. \tag{4.28}$$

We have yet to specify the local wave speeds s^l, s^r. Different choices of the speeds lead to different schemes; presently we describe three of the most important ones.

4.2.3. Lax–Friedrichs scheme. To ensure that waves from neighboring Riemann problems (4.24) do not interact, the maximum allowed wave speeds are

$$s_{j+1/2}^l = -\frac{\Delta x}{\Delta t}, \qquad s_{j+1/2}^r = \frac{\Delta x}{\Delta t}. \tag{4.29}$$

These wave speeds substituted in (4.27) lead to the Lax–Friedrichs flux

$$F_{j+1/2}^n = F^{\mathrm{LxF}}\big(U_j^n, U_{j+1}^n\big) = \frac{f(U_j^n) + f(U_{j+1}^n)}{2} - \frac{\Delta x}{2\Delta t}\left(U_{j+1}^n - U_j^n\right). \tag{4.30}$$

The Lax–Friedrichs scheme (4.14), (4.30) is very simple to implement. Numerical results for Burgers' equation with initial data (4.17) and (4.18) are shown in Figure 4.8 (compare to figure 4.4). The results show that the approximate solutions are stable and nonoscillatory and approximate the entropy solution, unlike the Roe scheme. However, the computed solutions are *diffusive*. The shocks are smeared to a considerable extent. The numerical results are inferior to those obtained with the Godunov scheme.

4.2.4. Rusanov scheme (1961). The Lax–Friedrichs scheme was quite diffusive around shocks. A possible explanation lies in the choice of the wave speeds (4.29). These speeds were the maximum allowed speeds and did not take into the account the speeds of propagation of the problem under consideration. A better, *locally selected*, choice of speeds is given by

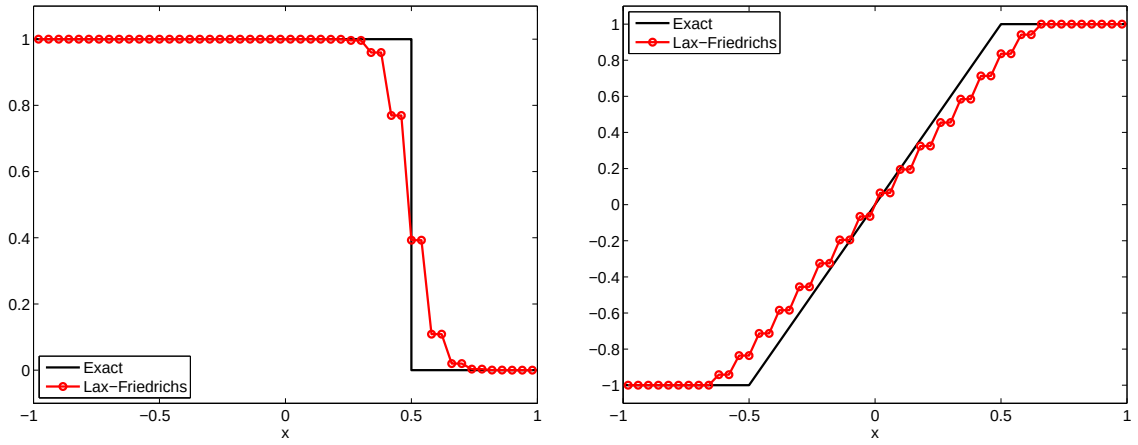
$$s_{j+1/2}^r = s_{j+1/2}, \quad s_{j+1/2}^l = -s_{j+1/2}, \tag{4.31}$$

where

$$s_{j+1/2} = \max\left(|f'(U_j^n)|,\ |f'(U_{j+1}^n)|\right).$$

The resulting flux (4.27), called the *Rusanov* (or *Local Lax–Friedrichs*) flux, is given by

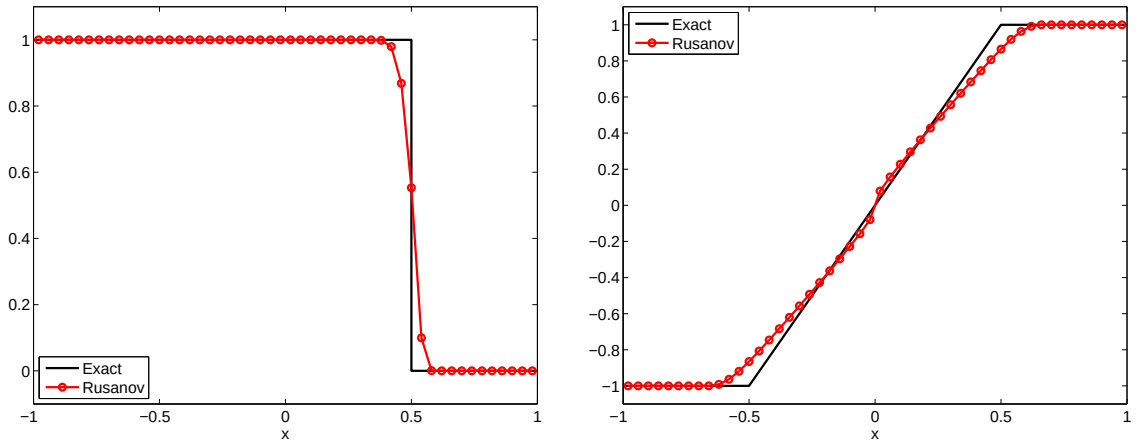
$$\begin{aligned} F_{j+1/2}^n &= F^{\mathrm{Rus}}\big(U_j^n, U_{j+1}^n\big) \\ &= \frac{f(U_j^n) + f(U_{j+1}^n)}{2} - \frac{\max\left(|f'(U_j^n)|,\ |f'(U_{j+1}^n)|\right)}{2}\left(U_{j+1}^n - U_j^n\right). \end{aligned} \tag{4.32}$$



(a) Shock solution with initial data (4.17) at $t = 1$.

(b) Rarefaction wave solution with initial data (4.18) at $t = 0.5$.

FIGURE 4.8. Approximate solution for Burgers' equation with the Lax–Friedrichs scheme with 50 mesh points. `[burgers_disc.m]`



(a) Shock solution with initial data (4.17) at $t = 1$.

(b) Rarefaction wave solution with initial data (4.18) at $t = 0.5$.

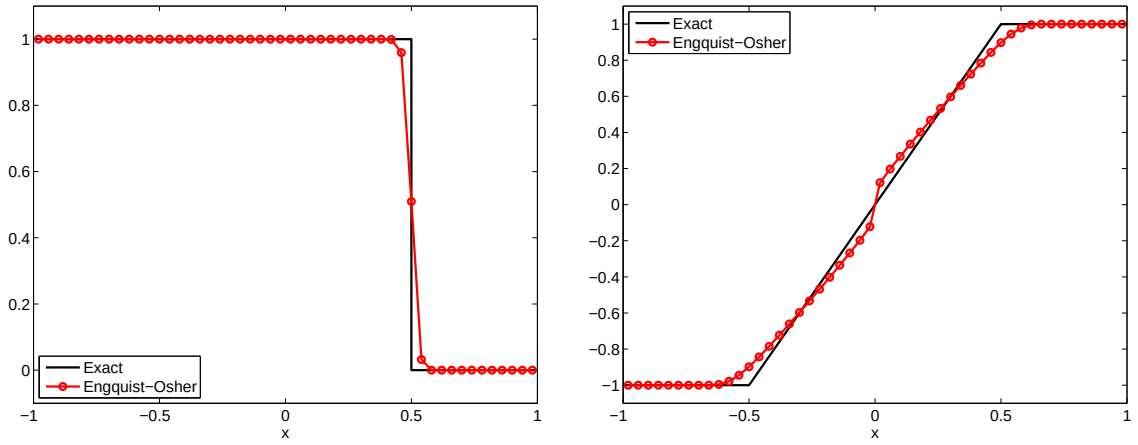
FIGURE 4.9. Approximate solution for Burgers' equation with the Rusanov scheme using 50 mesh points. `[burgers_disc.m]`

The Rusanov scheme (4.14), (4.32) leads to a considerable improvement in results over the Lax–Friedrichs scheme, as shown in Figure 4.9.

4.2.5. Engquist–Osher scheme. A related scheme is the Engquist–Osher scheme, which has flux

$$\begin{aligned} F^n_{j+1/2} &= F^{\mathrm{EO}}\left(U^n_j, U^n_{j+1}\right) \\ &= \frac{f(U^n_j) + f(U^n_{j+1})}{2} \; - \; \frac{1}{2}\int_{U^n_j}^{U^n_{j+1}} |f'(\theta)| \, d\theta. \end{aligned} \tag{4.33}$$

Although it is difficult to write the Engquist–Osher flux as an approximate Riemann solver, it shares several features of approximate Riemann solvers. When the flux function has a single minimum at a point ω and no maxima (which is the case for most convex functions), the Engquist–Osher flux can be



(a) Shock solution with initial data (4.17) at $t = 1$.

(b) Rarefaction wave solution with initial data (4.18) at $t = 0.5$.

FIGURE 4.10. Approximate solution for Burgers' equation with the Engquist–Osher scheme using 50 mesh points. `[burgers_disc.m]`

explicitly computed as

$$F^{\mathrm{EO}}\left(U_j^n, U_{j+1}^n\right) = f\left(\max\left(U_j^n, \omega\right)\right) + f\left(\min\left(U_{j+1}^n, \omega\right)\right) - f(\omega). \tag{4.34}$$

For convex fluxes with minimum at ω, we denote

$$f^+(U) = f\big(\max\left(U, \omega\right)\big), \quad f^-(U) = f\big(\min\left(U, \omega\right)\big), \tag{4.35}$$

as the *positive* (increasing) and *negative* (decreasing) parts of f. As only the flux difference appear in (4.14), we can neglect the constant term $f(\omega)$ in (4.34) and rewrite the Engquist–Osher scheme for convex fluxes as

$$F^{\mathrm{EO}}\left(U_j^n, U_{j+1}^n\right) = f^+\left(U_j^n\right) + f^-\left(U_{j+1}^n\right). \tag{4.36}$$

Hence, the Engquist–Osher scheme is a *flux splitting scheme*, as it separates the flux into its positive and negative parts and takes the direction of propagation into account.

Exercise 4.3. *Prove that the Engquist–Osher flux* (4.33) *can be written as* (4.34) *when the flux function has a single minimum at a point* ω.

4.3. Comparison of different finite volume schemes

We compare all the numerical fluxes presented in this section for two sets of initial data. First, we consider the initial data (4.17) and compare the different numerical fluxes for a mesh consisting of 50 mesh points in Figure 4.11. The results show that the Godunov, Roe and Engquist–Osher schemes agree in this case. In fact, simple calculations show that in this case these three fluxes are equivalent. The Godunov scheme is clearly more accurate than the Rusanov scheme. The Lax–Friedrichs scheme leads to the largest amount of error as it smears the shock wave. We perform a convergence study of the schemes and present the results in Figure 4.12 (a). The solutions for initial data (4.17) are computed on a sequence of meshes and the error (with respect to the exact solution) in L^1 is computed and plotted with respect to the number of mesh points (decreasing mesh sizes). The plot indicates that all the schemes converge as the mesh is refined. The convergence for the Godunov scheme is faster than the Rusanov and Lax–Friedrichs schemes, although the rate of convergence is similar for all the schemes. The results clearly show that the Godunov scheme is superior to the Rusanov and Lax–Friedrichs scheme. However, we must consider the fact that both the Lax–Friedrichs and Rusanov schemes have faster run times than the Godunov scheme. Hence, a fair comparison requires us to plot the *computational efficiency*. To do so, we compute with all the schemes on a sequence of meshes and plot the L^1 error with respect to the *runtime* for each scheme in Figure 4.12 (b). The figure shows the obvious: Decreasing the mesh size

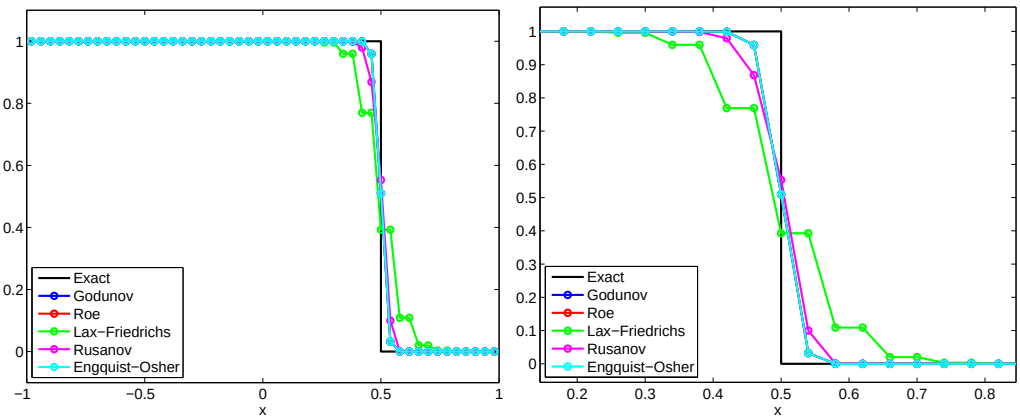
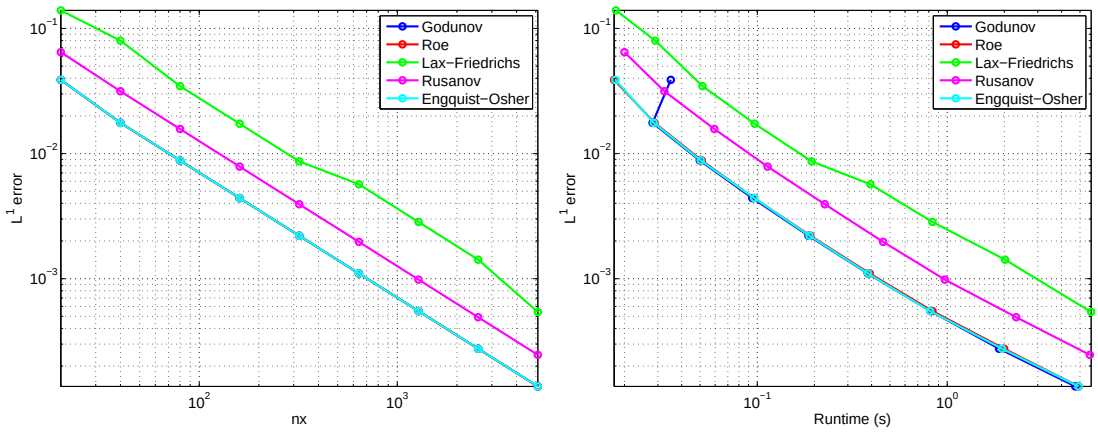
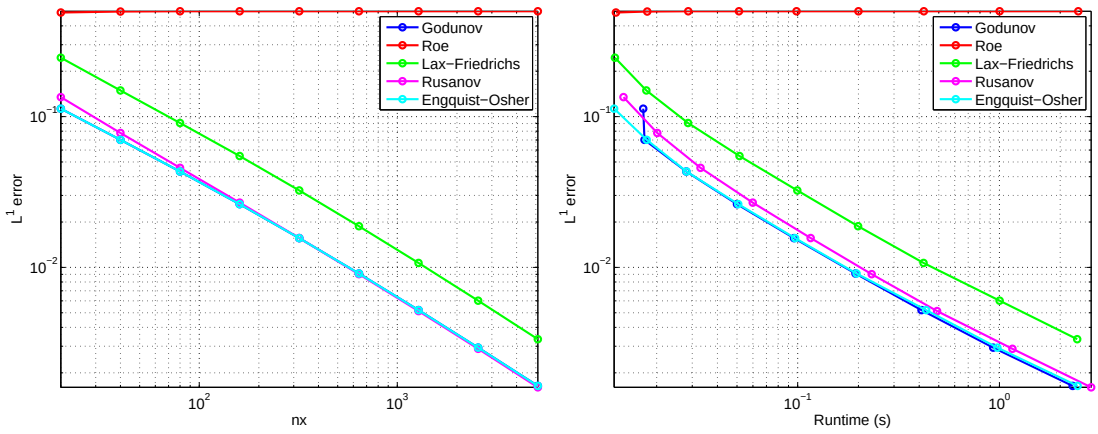


FIGURE 4.11. Approximate solution for Burgers' equation with the Godunov, Lax–Friedrichs, Rusanov, Roe and Engquist–Osher schemes at time $t = 1.5$ with 50 mesh points for initial data (4.17). `[burgers_disc.m]`



(a) L^1 error vs. number of mesh points

(b) L^1 error vs. runtime



(c) L^1 error vs. number of mesh points

(d) L^1 error vs. runtime

FIGURE 4.12. Convergence study for Burgers' equation with the Godunov, Lax–Friedrichs, Rusanov, Roe and Engquist–Osher schemes. Top row: initial data (4.17); bottom row: initial data (4.18). `[burgers_disc_error.m]`

gives more accurate approximations, but also leads to a higher run-time. The Enquist–Osher scheme turns out to be the most efficient in this case, at least for coarser meshes. We recall that the Godunov, Engquist–Osher and Roe schemes give the same numerical approximation on this problem. However, their run times are different. We point out that the schemes agree in terms of runtimes for highly refined meshes. Despite being the fastest on a given mesh, the Lax–Friedrichs continues to be the most inefficient scheme in this example.

Next, we repeat the experiments with the *rarefaction* initial data (4.18). The error vs. number of mesh points and error vs. run time is plotted in Figure 4.12 (c) and (d), respectively. The figures show that the Roe scheme does not converge in this case to the entropy solution. The Godunov and Engquist–Osher schemes are equivalent and lead to smaller errors than the Rusanov scheme, at least for coarse meshes. The Lax–Friedrichs scheme leads to the largest errors among the converging schemes. The computational efficiency plot shows that the Rusanov scheme is the most efficient in this case. Thus, the *optimal* scheme is a problem dependent concept.

4.4. Consistent, conservative and monotone schemes

The numerical results (particularly convergence results, Figures 4.12) show that most of the schemes of the previous section converge to the entropy solution of the scalar conservation law (4.1). However, the Roe scheme (4.23) may not converge to the entropy solution in some cases. It is easy to design schemes that lead to incorrect or unstable solutions. One example is the scheme with the flux function

$$F^n_{j+1/2} = F^{\text{Cen}}\big(U^n_j, U^n_{j+1}\big) = \frac{f(U^n_j) + f(U^n_{j+1})}{2}. \tag{4.37}$$

This flux gives the standard central difference scheme for the conservation law (4.1). As shown before, it ends up being unconditionally unstable, even for linear equations. Another possible flux is the one-sided flux

$$F^n_{j+1/2} = F(U^n_j, U^n_{j+1}) = f(U^n_j). \tag{4.38}$$

If we consider Burgers' equation with initial data (4.17), then the one-sided scheme (4.38) reduces to the Godunov scheme and will provide a good approximation of the solution. However, with initial data (4.18), the scheme (4.38) reduces to the Roe scheme and provides an incorrect approximation.

The above examples show that certain fluxes are stable whereas others are unstable. Furthermore, some fluxes converge to the correct entropy solutions whereas others converge to wrong solutions. The natural question is what the criteria are for the schemes to be stable and to converge to the entropy solutions of the scalar conservation law (4.1). We provide answers in this and the following sections.

First, we will identify certain features of the schemes presented so far.

4.4.1. Conservative schemes. A numerical scheme for approximating (4.1) can be written in the generic form

$$U^{n+1}_j = H\big(U^n_{j-p}, \dots, U^n_{j+p}\big) \tag{4.39}$$

for some update function H and constant $p \in \mathbb{N}$. The update function H depends on $2p+1$ points. This set of points $\{U^n_{j-p}, \dots, U^n_{j+p}\}$ is called the *stencil* of the scheme. All the schemes presented so far have $p = 1$ and are three-point schemes. More general $(2p+1)$-point schemes will be presented in the sequel.

Definition 4.4 (Conservative scheme)**.** *Ignoring boundary conditions, a numerical scheme of generic form* (4.39) *approximating* (4.1) *is* conservative *if it satisfies*

$$\sum_j U^{n+1}_j = \sum_j U^n_j \quad \textit{for all } n. \tag{4.40}$$

Conservation is a natural requirement from a scheme as the solutions to the continuous problem (4.1) are conservative, in the sense that the integral of the solution is preserved over time.

Theorem 4.5. *Assume that* $H(0, \dots, 0) = 0$. *Then* (4.39) *is conservative* if and only if *there exists a function* $F^n_{j+1/2} = F(U^n_{j-p+1}, \dots, U^n_{j+p})$ *such that* (4.39) *can be written in the finite volume form* (4.14).

Proof. Necessity is straightforward: if (4.39) can be written as (4.14), then

$$\begin{aligned}\sum_j U_j^{n+1} &= \sum_j U_j^n - \frac{\Delta t}{\Delta x}\sum_j \left(F^n_{j+1/2} - F^n_{j-1/2}\right)\\ &= \sum_j U_j^n,\end{aligned}$$

since the second sum is a telescoping sum.

For sufficiency, define

$$G(U_{-p},\dots,U_p) := \frac{\Delta x}{\Delta t}\left(U_0 - H(U_{-p},\dots,U_p)\right)$$

(we drop the time dependence for the moment). By conservation, we have

$$\sum_j G(U_{j-p},\dots,U_{j+p}) = 0.$$

We want to show that there is an $F_{j+1/2} = F(U_{j-p+1},U_{j+p})$ such that

$$F_{j+1/2} - F_{j-1/2} = G(U_{j-p},\dots,U_{j+p}). \tag{4.41}$$

If will suffice to construct $F_{j+1/2}$ for $j=0$. Since the assumption holds for *any* sequence, select first U_j such that $U_j = 0$ whenever $j \leqslant -p$ or $j > p$. Then

$$\begin{aligned}0 = &\sum_j G(U_{j-p},\dots,U_{j+p})\\ &= G(0,\dots,0,U_{-p+1}) + \cdots + G(0,U_{-p+1},\dots,U_p)\\ &\quad + G(U_{-p+1},\dots,U_p,0) + \cdots + G(U_p,0,\dots,0)\\ &= F_{1/2} + B\end{aligned}$$

where we have defined $F_{1/2} := G(0,\dots,0,U_{-p+1}) + \cdots + G(0,U_{-p+1},\dots,U_p)$ and $B := G(U_{-p+1},\dots,U_p,0) + \cdots + G(U_p,0,\dots,0)$. Next, select U_j such that $U_j = 0$ whenever $j < -p$ or $j > p$. Then

$$\begin{aligned}0 = &\sum_j G(U_{j-p},\dots,U_{j+p})\\ &= G(0,\dots,0,U_{-p}) + \cdots + G(0,U_{-p},\dots,U_{p-1}) + G(U_{-p},\dots,U_p)\\ &\quad + G(U_{-p+1},\dots,U_p,0) + \cdots + G(U_p,0,\dots,0)\\ &= F_{-1/2} + G(U_{-p},\dots,U_p) + B.\end{aligned}$$

Subtracting these two identities, we get the desired identity (4.41). □

The definition of conservation needs to be modified when boundaries are included in the discussion.

4.4.2. Consistent schemes. Another crucial requirement is consistency. Let some $(2p+1)$-point scheme (4.39) be given with numerical flux

$$F^n_{j+1/2} = F(U^n_{j-p+1},\dots,U^n_{j+p}). \tag{4.42}$$

Definition 4.6 (Consistency)**.** *A finite volume scheme* (4.14) *with numerical flux function* (4.42) *is* consistent *if*

$$F(U,\dots,U) = f(U) \quad \textit{for all } U \in \mathbb{R}. \tag{4.43}$$

It is straightforward to check that all the two-point numerical flux functions presented so far are consistent. Consistency is a natural requirement to ensure that the scheme approximates the correct conservation law.

Conservation and consistency are shared by all the schemes discussed until now. However, these criteria do not ensure either stability or convergence; for instance, the central scheme (4.37) is both consistent and conservative, but fails to be stable. Similarly, the Roe scheme (4.23) is consistent and conservative, but does not converge to the entropy solution in certain numerical experiments. We need another criterion.

4.4.3. Monotone schemes. Recall from Theorem 3.10 that the conservation law (4.1) is *monotonicity preserving*, meaning that if U and V are entropy solutions of (4.1) with initial data U_0 and V_0, respectively, then

$$U_0(x) \leqslant V_0(x) \quad \text{for all } x \quad \Rightarrow \quad U(x,t) \leqslant V(x,t) \quad \text{for all } x, t.$$

It is desirable for numerical schemes to posses this property. At the discrete level, it is defined by the following.

Definition 4.7. *A numerical scheme* (4.39) *is* monotone *if the update function H is non-decreasing in each of its arguments.*

If (4.39) can be written in the finite volume form (4.14) with a two-point flux function (so $p = 1$ in (4.42)), then a sufficient condition for monotonicity is given by

Lemma 4.8. *Consider the finite volume scheme* (4.14) *with a locally Lipschitz continuous two-point flux $F = F(a,b)$. Then the method* (4.14) *is monotone* if and only if

$$\begin{aligned} &a \mapsto F(a,b) \quad \textit{is non-decreasing for fixed } b, \\ &b \mapsto F(a,b) \quad \textit{is non-increasing for fixed } a, \end{aligned} \tag{4.44}$$

and the following CFL-type condition holds:

$$\left|\frac{\partial F}{\partial a}(v,w)\right| + \left|\frac{\partial F}{\partial b}(u,v)\right| \leqslant \frac{\Delta x}{\Delta t} \qquad \forall\ u,v,w. \tag{4.45}$$

The proof of this lemma is left as an exercise to the reader.

4.4.4. Examples of monotone schemes. Assume that the flux function f in (4.1) is a differentiable function. Then the following schemes are monotone:

- The Lax–Friedrichs scheme (4.30) has a smooth C^1 numerical flux function. Explicit computations result in
$$\begin{aligned} \frac{\partial F}{\partial a} &= \frac{1}{2}\left(f'(a) + \frac{\Delta x}{\Delta t}\right) \\ \frac{\partial F}{\partial b} &= \frac{1}{2}\left(f'(b) - \frac{\Delta x}{\Delta t}\right). \end{aligned}$$
The condition (4.45) then follows from the CFL condition (4.10).
- The Rusanov scheme (4.32) is monotone. The proof follows as above.
- The Engquist–Osher scheme (4.33) has a smooth numerical flux. A direct calculation shows that
$$\begin{aligned} \frac{\partial F}{\partial a} &= \frac{1}{2}\left(f'(a) + |f'(a)|\right) \\ \frac{\partial F}{\partial a} &= \frac{1}{2}\left(f'(b) - |f'(b)|\right). \end{aligned}$$
Furthermore, the CFL condition (4.45) is a consequence of (4.10). Hence, the Engquist–Osher scheme is monotone.
- The Godunov scheme (4.15) has a Lipschitz continuous flux function and is monotone. If e.g. $U_j \leqslant U_{j+1}$ then increasing U_j will shrink the interval over which the minimum in (4.15) is taken, while increasing U_{j+1} will expand it.

On the other hand, it is easy to check that the central flux (4.37), the one-sided flux (4.38) and the Roe flux (4.23) are not monotone. Thus, monotonicity of the scheme provides a demarcation between robust and potentially non-robust schemes.

4.5. Stability properties of monotone schemes

The numerical experiments indicate that monotone schemes are stable and converge to the entropy solution, whereas the non-monotone schemes may be unstable or may converge to the incorrect solution. A crucial step in proving convergence is to obtain stability estimates on the approximate solutions, computed by the schemes.

From our study of the continuous problem in the previous section, the entropy solutions of the continuous problem (4.1) satisfy the following stability estimates:

(i) L^∞ estimate (3.41) as a consequence of a maximum principle
(ii) L^p estimates (3.30) for $1 \leqslant p < \infty$ as a consequence of the entropy inequality (3.29)
(iii) The TV estimate (3.42) showing that the solutions are TVD
(iv) The time continuity estimate (3.43).

The central philosophy of numerical analysis is to devise numerical schemes that preserve stability properties of the underlying continuous problem. Hence, it is natural to examine whether the finite volume schemes presented in this section satisfy discrete versions of the above stability estimates.

4.5.1. L^∞ bounds. A consistent three point finite volume scheme (4.14) can be written as

$$U_j^{n+1} = H(U_{j-1}^n, U_j^n, U_{j+1}^n), \tag{4.46}$$

with the consistency of the fluxes F in (4.14) implying that

$$H(U, U, U) = U.$$

The scheme (4.46) is monotone if H is non-decreasing in each of its three arguments. The Godunov, Engquist–Osher, Lax–Friedrichs and Rusanov schemes (under a CFL condition like (4.10)) are examples of three-point monotone schemes. These schemes satisfy the following *discrete maximum principle*.

Lemma 4.9. *Let U_j^n be the approximate solutions generated by a three-point consistent monotone scheme of the form* (4.46)*. Then the solutions satisfy*

$$\min\left(U_{j-1}^n,\ U_j^n,\ U_{j+1}^n\right) \leqslant U_j^{n+1} \leqslant \max\left(U_{j-1}^n,\ U_j^n,\ U_{j+1}^n\right) \quad \text{for all } n, j. \tag{4.47}$$

In particular, we have

$$\min_i U_i^0 \leqslant U_j^n \leqslant \max_i U_i^0 \qquad \text{for all } n, j. \tag{4.48}$$

Proof. Let $\bar{U}_j^n = \max\left(U_{j-1}^n, U_j^n, U_{j+1}^n\right)$. By definition,

$$U_{j-1}^n, U_j^n, U_{j+1}^n \leqslant \bar{U}_j^n.$$

Since the scheme (4.46) is monotone, the update function H is monotone non-decreasing in each of its arguments. Therefore

$$\begin{aligned}
U_j^{n+1} = H(U_{j-1}^n, U_j^n, U_{j+1}^n) &\leqslant H(\bar{U}_j^n, U_j^n, U_{j+1}^n) && \text{(monotonicity)}\\
&\leqslant H(\bar{U}_j^n, \bar{U}_j^n, U_{j+1}^n) && \text{(monotonicity)}\\
&\leqslant H(\bar{U}_j^n, \bar{U}_j^n, \bar{U}_j^n) && \text{(monotonicity)}\\
&= \bar{U}_j^n && \text{(consistency)}.
\end{aligned}$$

The minimum principle follows analogously. Iterating the maximum principle (4.47) over all time levels up to t^n yields the L^∞ bound (4.48). □

4.5.2. Entropy inequalities and L^p bounds. The key to obtaining L^p bounds and characterizing the correct entropy solution is to obtain a discrete version of the entropy inequality (3.29). As we have seen in Chapter 3, it is enough to obtain a discrete version of the Kruzkhov entropy inequality (3.32). We proceed to do so below.

For a constant $k \in \mathbb{R}$, define the Crandall–Majda numerical entropy flux as

$$\begin{aligned}
Q_{j+1/2}^n &= Q(U_j^n, U_{j+1}^n)\\
&= F(U_j^n \vee k, U_{j+1}^n \vee k) - F(U_j^n \wedge k, U_{j+1}^n \wedge k),
\end{aligned} \tag{4.49}$$

where we use the notation

$$a \vee b = \max(a, b), \qquad a \wedge b = \min(a, b).$$

Note that this numerical entropy flux is consistent with the Kruzkhov entropy flux (3.31) whenever the numerical flux F is consistent, as

$$\begin{aligned} Q(U,U) &= F(U \vee k, U \vee k) - F(U \wedge k, U \wedge k) \\ &= f(U \vee k) - f(U \wedge k) \\ &= \operatorname{sign}(U-k)(f(U) - f(k)) \\ &= q(U,k). \end{aligned}$$

In the following lemma we obtain a discrete version of (3.32).

Lemma 4.10 (Crandall–Majda [**CM80**])**.** *Let U_j^n be an approximate solution computed by a consistent, conservative and monotone three-point scheme* (4.46)*. Then U satisfies the discrete entropy inequality*

$$|U_j^{n+1} - k| - |U_j^n - k| + \frac{\Delta t}{\Delta x}\left(Q_{j+1/2}^n - Q_{j-1/2}^n\right) \leqslant 0 \tag{4.50}$$

for all n, j. In particular, if $U_0 \in L^1(\mathbb{R})$ then

$$\sum_j |U_j^n| \Delta x \leqslant \|U_0\|_{L^1(\mathbb{R})} \qquad \forall\ n \in \mathbb{N}. \tag{4.51}$$

Proof. By definition of the scheme (4.46) and (4.14), we have

$$\begin{aligned} &H\left(U_{j-1}^n \vee k, U_j^n \vee k, U_{j+1}^n \vee k\right) \\ &\qquad = U_j^n \vee k - \frac{\Delta t}{\Delta x}\left(F\left(U_j^n \vee k, U_{j+1}^n \vee k\right) - F\left(U_{j-1}^n \vee k, U_j^n \vee k\right)\right) \end{aligned}$$

and

$$\begin{aligned} &H\left(U_{j-1}^n \wedge k, U_j^n \wedge k, U_{j+1}^n \wedge k\right) \\ &\qquad = U_j^n \wedge k - \frac{\Delta t}{\Delta x}\left(F\left(U_j^n \wedge k, U_{j+1}^n \wedge k\right) - F\left(U_{j-1}^n \wedge k, U_j^n \wedge k\right)\right). \end{aligned}$$

Using the definition of $Q_{j+1/2}^n$, we obtain

$$\begin{aligned} &H\left(U_{j-1}^n \vee k, U_j^n \vee k, U_{j+1}^n \vee k\right) - H\left(U_{j-1}^n \wedge k, U_j^n \wedge k, U_{j+1}^n \wedge k\right) \\ &\qquad = U_j^n \vee k - U_j^n \wedge k - \frac{\Delta t}{\Delta x}\left(Q_{j+1/2}^n - Q_{j-1/2}^n\right) \\ &\qquad = |U_j^n - k| - \frac{\Delta t}{\Delta x}\left(Q_{j+1/2}^n - Q_{j-1/2}^n\right). \end{aligned} \tag{4.52}$$

By monotonicity and the definition of H (4.46),

$$\begin{aligned} H\left(U_{j-1}^n \vee k, U_j^n \vee k, U_{j+1}^n \vee k\right) &\geqslant H\left(U_{j-1}^n, U_j^n, U_{j+1}^n\right) = U_j^{n+1}, \\ H\left(U_{j-1}^n \wedge k, U_j^n \wedge k, U_{j+1}^n \wedge k\right) &\leqslant H\left(U_{j-1}^n, U_j^n, U_{j+1}^n\right) = U_j^{n+1}. \end{aligned} \tag{4.53}$$

Similarly, by monotonicity and consistency of H,

$$\begin{aligned} H\left(U_{j-1}^n \vee k, U_j^n \vee k, U_{j+1}^n \vee k\right) &\geqslant H(k,k,k) = k, \\ H\left(U_{j-1}^n \wedge k, U_j^n \wedge k, U_{j+1}^n \wedge k\right) &\leqslant H(k,k,k) = k. \end{aligned} \tag{4.54}$$

By subtracting (4.54) from (4.53) and using (4.52), we obtain

$$|U_j^n - k| - \frac{\Delta t}{\Delta x}\left(Q_{j+1/2}^n - Q_{j-1/2}^n\right) \geqslant |U_j^{n+1} - k|,$$

which proves (4.50). Setting $k = 0$ and summing (4.50) over j and n gives (4.51). □

Remark 4.11. *The discrete entropy inequality* (4.50) *can be used with an approximation argument for Kruzkhov entropies* (3.31) *to obtain discrete entropy inequalities for any convex function q. This implies L^p bounds on the solution by summing over all mesh points.*

4.5.3. TV bounds. Another essential stability estimate for scalar conservation laws (4.1) is the BV estimate (3.42), which provides control over the oscillations in the solution. This estimate says that the entropy solutions of (4.1) are TVD, meaning that the TV norm does not increase in time. We need to obtain a discrete version of this total variation. The total variation of a piecewise constant function at time level t^n is given by

$$\|U^n\|_{TV(\mathbb{R})} = \sum_j |U^n_{j+1} - U^n_j|.$$

The first step for obtaining a TV bound is rewriting the finite volume scheme in the *incremental form*

$$U^{n+1}_j = U^n_j + C^n_{j+1/2}\left(U^n_{j+1} - U^n_j\right) - D^n_{j-1/2}\left(U^n_j - U^n_{j-1}\right) \tag{4.55}$$

with incremental coefficients

$$C^n_{j+1/2} = \frac{\Delta t}{\Delta x}\frac{f(U^n_j) - F^n_{j+1/2}}{U^n_{j+1} - U^n_j}, \quad D^n_{j+1/2} = \frac{\Delta t}{\Delta x}\frac{f(U^n_{j+1}) - F^n_{j+1/2}}{U^n_{j+1} - U^n_j}. \tag{4.56}$$

A straightforward calculation shows that the incremental form (4.55) is equivalent to the standard finite volume form (4.14). The advantage of the incremental form lies in the following useful lemma.

Lemma 4.12 (Harten's Lemma [**Har83**])**.** *Consider the scheme* (4.55)*.*

(i) If the coefficients satisfy

$$C^n_{j+1/2},\ D^n_{j+1/2} \geqslant 0 \quad \text{and} \quad C^n_{j+1/2} + D^n_{j+1/2} \leqslant 1 \quad \text{for all } n, j \tag{4.57}$$

then solutions computed with (4.55) *are TVD, i.e. they satisfy*

$$\sum_j |U^{n+1}_{j+1} - U^{n+1}_j| \leqslant \sum_j |U^n_{j+1} - U^n_j| \qquad \text{for all } n. \tag{4.58}$$

(ii) If the coefficients satisfy

$$C^n_{j+1/2},\ D^n_{j+1/2} \geqslant 0 \quad \text{and} \quad C^n_{j+1/2} + D^n_{j-1/2} \leqslant 1 \quad \text{for all } n, j \tag{4.59}$$

then $\|U^{n+1}\|_{L^\infty} \leqslant \|U^n\|_{L^\infty}$ *for all* n*.*

Proof. Using the incremental form (4.55), we obtain

$$\begin{aligned} U^{n+1}_{j+1} - U^{n+1}_j = &\left(1 - C^n_{j+1/2} - D^n_{j+1/2}\right)\left(U^n_{j+1} - U^n_j\right) \\ &+ C^n_{j+3/2}\left(U^n_{j+2} - U^n_{j+1}\right) + D^n_{j-1/2}\left(U^n_j - U^n_{j-1}\right). \end{aligned}$$

Taking absolute values on both sides of the above equation and using (4.57) yields

$$\begin{aligned} \left|U^{n+1}_{j+1} - U^{n+1}_j\right| \leqslant &\left(1 - C^n_{j+1/2} - D^n_{j+1/2}\right)\left|U^n_{j+1} - U^n_j\right| \\ &+ C^n_{j+3/2}\left|U^n_{j+2} - U^n_{j+1}\right| + D^n_{j-1/2}\left|U^n_j - U^n_{j-1}\right|. \end{aligned}$$

Summing over j and identifying equal terms, we obtain the TVD estimate (4.58).

For the L^∞ bound, rewrite U^{n+1}_j as

$$U^{n+1}_j = C^n_{j+1/2}U^n_{j+1} + \left(1 - C^n_{j+1/2} - D^n_{j-1/2}\right)U^n_j + D^n_{j-1/2}U^n_{j-1}.$$

The condition (4.59) ensures that U^{n+1}_j is a convex combination of U^n_{j+1}, U^n_j and U^n_{j-1}, which proves the L^∞ bound. □

Harten's lemma allows us to check whether a scheme in incremental form is TVD or not. It turns out that consistent monotone schemes are TVD.

Lemma 4.13. *Any consistent and conservative monotone three-point finite volume scheme* (4.14) *is TVD under the CFL condition* (4.45)*, i.e., it satisfies* (4.58)*.*

Proof. We need to check if the scheme satisfies the conditions (4.57) of Harten's lemma. By explicit computations we obtain

$$\begin{aligned} C^n_{j+1/2} &= \frac{\Delta t}{\Delta x}\frac{f(U^n_j) - F(U^n_j, U^n_{j+1})}{U^n_{j+1} - U^n_j} && \text{(definition (4.56))} \\ &= \frac{\Delta t}{\Delta x}\frac{F(U^n_j, U^n_j) - F(U^n_j, U^n_{j+1})}{U^n_{j+1} - U^n_j} && \text{(consistency)} \\ &\geqslant 0 && \text{(monotonicity).} \end{aligned}$$

The correct sign of D can be similarly checked. We also have

$$\begin{aligned} C^n_{j+1/2} &= \frac{\Delta t}{\Delta x}\frac{f(U^n_j) - F(U^n_j, U^n_{j+1})}{U^n_{j+1} - U^n_j} && \text{(definition (4.56))} \\ &= \frac{\Delta t}{\Delta x}\frac{F(U^n_j, U^n_j) - F(U^n_j, U^n_{j+1})}{U^n_{j+1} - U^n_j} && \text{(consistency)} \\ &\leqslant \frac{\Delta t}{\Delta x}\left|\frac{\partial F}{\partial b}\right| && \text{(Lipschitz flux).} \end{aligned}$$

Similarly,

$$D^n_{j+1/2} \leqslant \frac{\Delta t}{\Delta x}\left|\frac{\partial F}{\partial a}\right|.$$

Therefore,

$$\begin{aligned} 1 - C^n_{j+1/2} - D^n_{j+1/2} &\geqslant 1 - \frac{\Delta t}{\Delta x}\left(\left|\frac{\partial F}{\partial b}\right| + \left|\frac{\partial F}{\partial a}\right|\right) \\ &\geqslant 0 \end{aligned}$$

by the CFL condition (4.45). □

4.5.4. Time continuity. The entropy solution of (4.1) satisfies the time continuity property

$$\|U(t) - U(s)\|_{L^1(\mathbb{R})} \leqslant |t-s|\|f\|_{\text{Lip}}\|U_0\|_{TV(\mathbb{R})},$$

where $\|f\|_{\text{Lip}}$ is the Lipschitz constant of f over the range of U. We can easily prove a similar estimate for conservative and consistent schemes.

Lemma 4.14. *Let $\{U^n_j\}$ be computed with a consistent and conservative monotone scheme (4.1) with a Lipschitz continuous numerical flux function $F = F(a,b)$. Then $\{U^n_j\}$ satisfies the time continuity property*

$$\sum_{j\in\mathbb{Z}} \left|U^n_j - U^m_j\right|\Delta x \leqslant |t^n - t^m| C_F \|U_0\|_{TV(\mathbb{R})} \tag{4.60}$$

where C_F is the Lipschitz constant of F.

Proof. Take absolute values of (4.14) and sum over all $j \in \mathbb{Z}$ to obtain

$$\Delta x \sum_j \left|U^{n+1}_j - U^n_j\right| = \Delta t \sum_j \left|F^n_{j+1/2} - F^n_{j-1/2}\right|$$

(add and subtract $F(U^n_j, U^n_j)$ and apply the triangle inequality)

$$\leqslant \Delta t \sum_j \left|F(U^n_j, U^n_{j+1}) - F(U^n_j, U^n_j)\right| + \frac{\Delta t}{\Delta x}\sum_j \left|F(U^n_j, U^n_j) - F(U^n_{j-1}, U^n_j)\right|$$

(use the intermediate value theorem)

$$\leqslant \Delta t \sum_j \left\|\frac{\partial F}{\partial a}\right\|_{L^\infty} \left|U^n_{j+1} - U^n_j\right| + \frac{\Delta t}{\Delta x}\sum_j \left\|\frac{\partial F}{\partial b}\right\|_{L^\infty} \left|U^n_j - U^n_{j-1}\right|$$

(apply Lipschitz continuity of F)

$$\begin{aligned}&\leqslant \Delta t C_F \sum_j \left|U_{j+1}^n - U_j^n\right| \\ &= \Delta t C_F \|U^n\|_{TV(\mathbb{R})}.\end{aligned}$$

Summing over $n, \dots, m$ and applying the TVD property from Lemma 4.13 yields the desired result. □

4.6. Convergence of monotone methods

In the previous sections, we have established that monotone consistent conservative schemes are stable in L^∞, are TVD and satisfy a discrete entropy condition (4.50). These stability estimates pave the way for showing that the approximate solutions computed by any monotone consistent conservative scheme converge to the entropy solution of (4.1). The following famous theorem, due to Lax and Wendroff, says that *if* the approximate solutions U_j^n computed by a consistent and conservative scheme converge, then the limit is a weak solution of the conservation law. For ease of notation, we will identify the approximate solutions with a piecewise constant function,

$$U^{\Delta x}(x,t) = U_j^n \qquad \text{for } (x,t) \in [x_{j-1/2}, x_{j+1/2}) \times [t^n, t^{n+1}). \tag{4.61}$$

Theorem 4.15 (Lax–Wendroff)**.** *Let U_j^n be approximate solutions of* (4.1)*, computed by a conservative and consistent finite volume scheme* (4.14) *with a differentiable (or Lipschitz) numerical flux function F, where U_j^0 is given by* (4.4)*. Assume that $U_0 \in L^\infty(\mathbb{R})$ and that the approximating functions $U^{\Delta x}$*

- *are uniformly bounded, i.e,*

$$\|U^{\Delta x}\|_{L^\infty(\mathbb{R}\times\mathbb{R}_+)} \leqslant C \qquad \forall\ \Delta x > 0 \tag{4.62}$$

  for some constant $C > 0$;
- *converge in $L^1_{\text{loc}}(\mathbb{R}\times\mathbb{R}_+)$ as $\Delta x, \Delta t \to 0$ to some function U.*

Then U is a weak solution of (4.1)*, with initial data U_0.*

Proof. Let $\varphi \in C_c^\infty(\mathbb{R}\times[0,\infty))$ be a given test function and denote $\varphi_j^n = \varphi(x_j, t^n)$. By multiplying (4.14) by $\Delta x \varphi_j^n$ and summing over $j \in \mathbb{Z}$ and $n \in \mathbb{N}_0$, we obtain

$$0 = \Delta x \Delta t \sum_{j=-\infty}^{\infty} \sum_{n=0}^{\infty} \left(\frac{U_j^{n+1} - U_j^n}{\Delta t} + \frac{F_{j+1/2}^n - F_{j-1/2}^n}{\Delta x} \right) \varphi_j^n$$

(summation by parts)

$$= -\Delta x \sum_{j=-\infty}^{\infty} U_j^0 \varphi_j^0 - \Delta x \Delta t \sum_{j=-\infty}^{\infty} \sum_{n=0}^{\infty} \left(U_j^{n+1} \frac{\varphi_j^{n+1} - \varphi_j^n}{\Delta t} + F_{j+1/2}^n \frac{\varphi_{j+1}^n - \varphi_j^n}{\Delta x} \right).$$

Note that since φ has compact support, only finitely many of the summands in the above expression are nonzero, and so all the sums are well-defined. As with $U^{\Delta x}$, define $\varphi^{\Delta x}(x,t) = \varphi_j^n$ for $(x,t) \in [x_{j-1/2}, x_{j+1/2}) \times [t^n, t^{n+1})$. Recognizing the above expression as a Riemann sum of step functions, it can be written as

$$\begin{aligned}&\int_{\mathbb{R}} U^{\Delta x}(x,0)\varphi^{\Delta x}(x,0)\,dx + \int_{\mathbb{R}}\int_0^\infty U^{\Delta x}(x,t+\Delta t)\frac{\varphi^{\Delta x}(x,t+\Delta t) - \varphi^{\Delta x}(x,t)}{\Delta t} dt\,dx \\ &+ \int_{\mathbb{R}}\int_0^\infty F\left(U^{\Delta x}(x,t), U^{\Delta x}(x+\Delta x,t)\right) \frac{\varphi^{\Delta x}(x+\Delta x,t) - \varphi^{\Delta x}(x,t)}{\Delta x} dt\,dx = 0.\end{aligned} \tag{4.63}$$

Both $U^{\Delta x}$ and $\varphi^{\Delta x}$ converge boundedly a.e. to U and φ, respectively, so by Lebesgue's dominated convergence theorem, the first two terms converge to

$$\int_{\mathbb{R}} U(x,0)\varphi(x,0)\,dx + \int_{\mathbb{R}}\int_0^\infty U(x,t)\varphi_t(x,t) dt\,dx$$

as $\Delta x, \Delta t \to 0$. It is slightly more tricky to pass to the limit in the third term in (4.63), as $U^{\Delta x}$ occurs inside the (possibly nonlinear) function F. However, using (4.62) and the fact that F is differentiable, there is some $\widetilde{C} > 0$ such that

$$|F\left(U^{\Delta x}(x,t), U^{\Delta x}(x+\Delta x,t)\right) - F\left(U^{\Delta x}(x,t), U^{\Delta x}(x,t)\right)| \leqslant \widetilde{C}\left|U^{\Delta x}(x+\Delta x,t) - U^{\Delta x}(x,t)\right|.$$

Furthermore, by the consistency of F, we have $F\left(U^{\Delta x}(x,t), U^{\Delta x}(x,t)\right) = f(U^{\Delta x}(x,t))$. Adding and subtracting $f(U^{\Delta x}(x,t))$, we have

$$\begin{aligned}
&\int_{\mathbb{R}}\int_0^\infty F\left(U^{\Delta x}(x,t), U^{\Delta x}(x+\Delta x,t)\right)\frac{\varphi^{\Delta x}(x+\Delta x,t)-\varphi^{\Delta x}(x,t)}{\Delta x}\,dx\\
&=\int_{\mathbb{R}}\int_0^\infty f\left(U^{\Delta x}(x,t)\right)\frac{\varphi^{\Delta x}(x+\Delta x,t)-\varphi^{\Delta x}(x,t)}{\Delta x}\,dt\,dx\\
&\quad+\int_{\mathbb{R}}\int_0^\infty \Big(F\left(U^{\Delta x}(x,t), U^{\Delta x}(x+\Delta x,t)\right) - f\left(U^{\Delta x}(x,t)\right)\Big)\frac{\varphi^{\Delta x}(x+\Delta x,t)-\varphi^{\Delta x}(x,t)}{\Delta x}\,dt\,dx.
\end{aligned}$$

The first term converges towards $\int_{\mathbb{R}}\int_0^\infty f\left(U(x,t)\right)\varphi_x(x,t)\,dt\,dx$, while the second term is bounded by

$$\widetilde{C}\|\varphi_x\|_{L^\infty}\int_{\mathbb{R}}\int_0^\infty \left|U^{\Delta x}(x+\Delta x,t) - U^{\Delta x}(x,t)\right|\,dt\,dx,$$

which through approximation by C^∞ functions may be shown to converge to zero. In conclusion, the left-hand side of (4.63) converges to

$$\int_{\mathbb{R}}\int_0^\infty U\varphi_t + f(U)\varphi_x\,dt\,dx + \int_{\mathbb{R}} U(x,0)\varphi(x,0)\,dx,$$

thus proving that U is a weak solution. □

By applying the same technique as for the Lax–Wendroff theorem, we can easily prove the following result.

Lemma 4.16. *Assume that the conditions of the Lax–Wendroff theorem hold, and in addition that the discrete entropy inequality* (4.50) *holds. Then $U = \lim_{\Delta x\to 0} U^{\Delta x}$ is the entropy solution of* (3.4).

By the above lemma, all that remains in order to prove convergence to the entropy solution is to show that the sequence of computed solutions actually converge to some function U. For the sake of completeness we carry out the full proof in the remainder of this section. Although the details are somewhat technical, the main idea is as follows: The uniform L^1 and TV bounds (4.51) and (4.58), together with the time continuity (4.60), is enough to guarantee the existence of a convergent subsequence $U^{\Delta x'} \to U$. The uniqueness of the entropy solution U then implies that the *whole* sequence $U^{\Delta x}$ converges.

To carry out the above strategy we use the following two theorems from functional analysis.

Theorem 4.17 (Ascoli's theorem)**.** *Let (X, d_X) be a metric space and let $K \subset X$ be a relatively compact subset. Let $u_k : [0,T] \to K$ be a sequence of functions which are uniformly Lipschitz, i.e. there exists a constant $C > 0$ such that*

$$d_X(u_k(t), u_k(s)) \leqslant C|t-s| \qquad \forall\ k \in \mathbb{N}\ \forall\ t,s \in [0,T]. \tag{4.64}$$

Then there exists a subsequence $k(l)$ and a Lipschitz continuous function $u : [0,T] \to X$ such that $u_{k(l)} \to u$ uniformly as $l \to \infty$.

(Recall that a relatively compact set is a set whose closure is compact). The above version of Ascoli's theorem is a straightforward generalization of the standard Ascoli theorem, and the proof is very similar to the result for real-valued functions (see [**DS88**, p. 382]).

Theorem 4.18 (Helly's theorem)**.** *Let $[a,b]$ be a bounded interval and let $K \subset L^1([a,b])$. If there is an $M > 0$ such that*

$$\sup_{U\in K}\|U\|_{BV} \leqslant M \tag{4.65}$$

then K is relatively compact in $L^1([a,b])$. If $K \subset L^1(\mathbb{R})$ and, in addition to (4.65)*,*

$$\lim_{R\to\infty}\sup_{U\in K}\int_{\mathbb{R}\setminus[-R,R]}|U(x)|\,dx = 0 \tag{4.66}$$

then K is relatively compact in $L^1(\mathbb{R})$.

(Recall that the BV norm is defined as $\|U\|_{BV} = \|U\|_{L^1} + \|U\|_{TV}$.) The first part of Theorem 4.18 is what is usually referred to as Helly's theorem (see [**Giu84**, Theorem 1.19]). The *uniform integrability* assumption (4.66) is a standard assumption in order to go from convergence in $L^1_{\text{loc}}(\mathbb{R})$ to convergence in $L^1(\mathbb{R})$.

We are now ready to state the main convergence theorem of monotone schemes for scalar conservation laws.

Theorem 4.19. *Consider the scalar conservation law* (4.1) *with $f \in C^1(\mathbb{R})$ and $U_0 \in BV(\mathbb{R})$. Consider a consistent, conservative and monotone finite volume method* (4.14) *with a locally Lipschitz continuous flux F. Assume, in addition to the CFL condition* (4.45)*, that there is some $c > 0$ such that*

$$\frac{\Delta t}{\Delta x} \geqslant c. \tag{4.67}$$

Define the piecewise linear function

$$U^{\Delta x}(x,t) = \frac{t^{n+1} - t}{\Delta t} U_j^n + \frac{t - t^n}{\Delta t} U_j^{n+1} \qquad \text{for } x \in \mathcal{C}_j,\ t \in [t^n, t^{n+1}).$$

Then $U^{\Delta x} \to U$ in $L^1(\mathbb{R} \times [0,T])$ as $\Delta x, \Delta t \to 0$ for any $T > 0$, where U is the entropy solution of (3.4)

Proof. We only need to prove that the sequence $U^{\Delta x}$ converges to some function U, because the Lax–Wendroff-type Lemma 4.16 guarantees that the limit U is the entropy solution. (The fact that the function $U^{\Delta x}$ defined above is piecewise linear and not piecewise constant, as in (4.61), is merely out of convenience and does not change the validity of the Lax–Wendroff theorem.) Define

$$K = \big\{ U = U^{\Delta x}(\cdot, t) \ : \ t \in [0,T], \Delta x > 0 \big\},$$

i.e. the set of all functions $U : \mathbb{R} \to \mathbb{R}$ attained at some time by the numerical scheme. By the Crandall–Majda lemma 4.10 and the TVD bound in Lemma 4.13, the set K satisfies (4.65) with $M = \|U_0\|_{BV}$.

To show (4.66), let $\varepsilon > 0$ and let $r > 0$ be such that both $|U_0(x)| \leqslant \varepsilon$ for all $|x| > r$ and $\int_{\mathbb{R}\setminus[-r,r]} |U_0(x)|\,dx < \varepsilon$. Recall that monotonicity implies that

$$\min\big(U_{j-1}^n,\ U_j^n,\ U_{j+1}^n\big) \leqslant U_j^{n+1} \leqslant \max\big(U_{j-1}^n,\ U_j^n,\ U_{j+1}^n\big) \quad \text{for all } n, j.$$

By iterating over $t_0, \dots, t^N = T$ and using (4.67), we find that

$$|U^{\Delta x}(x,t)| \leqslant \varepsilon \qquad \text{for any } |x| > R,\ t \in [0,T] \tag{4.68}$$

where $R = r + \frac{T}{c}$. Let $J \in \mathbb{N}$ be such that $R \in \mathcal{C}_J$. By summing the discrete entropy inequality (4.50) with $k = 0$ over $|j| > R$ and $t^0, \dots, t^N$, we find that

$$\int_{\mathbb{R}\setminus[-R,R]} |U^{\Delta x}(x,t)|\,dx \leqslant \int_{\mathbb{R}\setminus[-R,R]} |U_0(x)|\,dx + \Delta t \sum_{n=0}^{N} |Q^n_{J+1/2}| + |Q^n_{-J-1/2}|.$$

By the definition of the Crandall–Majda numerical entropy flux Q, we can bound $|Q^n_{J+1/2}| \leqslant C_F(|U^n_J| + |U^n_{J+1}|) \leqslant 2C_F\varepsilon$, where C_F is the Lipschitz constant of F. Continuing from above, we see that

$$\int_{\mathbb{R}\setminus[-R,R]} |U^{\Delta x}(x,t)|\,dx \leqslant \varepsilon + \Delta t \sum_{n=0}^{N} 4C_F\varepsilon = \varepsilon(1 + 4TC_F).$$

The right-hand side is independent of t and Δx, so it follows that K also satisfies the uniform integrability condition (4.66).

Helly's theorem now implies that K is a (relatively) compact subset of $L^1(\mathbb{R})$. Viewing the computed solutions $U^{\Delta x}$ as functions from $[0,T]$ into K, the time continuity bound (4.60) allows us to apply Ascoli's theorem, yielding a subsequence $\Delta x' \to 0$ and a $U \in L^1(\mathbb{R} \times [0,T])$ such that $U^{\Delta x'} \to U$. The Lax–Wendroff theorem implies that U is the entropy solution. But the entropy solution is unique, so *any* convergent subsequence of $U^{\Delta x}$ has to converge to U. It follows that the *whole* sequence $U^{\Delta x}$ converges to U. □

4.7. A note on boundary conditions

The discussion so far has ignored boundary conditions. However, we need to specify boundary conditions as

- The problem itself may be an initial- boundary value problem and physical boundary conditions such as Dirichlet, Neumann or periodic boundary conditions might be specified.
- For an initial value problem on the whole real line, we need to truncate the domain, as a computational domain must be bounded. In this case, we need artificial or numerical boundary conditions.

Without exploring this issue in detail in the current discussion, we provide a recipe for implementing boundary conditions with finite volume schemes (4.14). Let $[x_L, x_R]$ be the physical or computational domain. Denote $x_{1/2} = x_L$ and $x_{N+1/2} = x_R$ (for a mesh with $N+1$ mesh points). We need the following *ghost cells*:

$$\mathcal{C}_0 = [x_L - \Delta x, x_L)\,, \quad \mathcal{C}_{N+1} = [x_R, x_R + \Delta x)\,. \tag{4.69}$$

We denote the cell averages of the unknown over $\mathcal{C}_0$ and $\mathcal{C}_{N+1}$ at time level t^n as U_0^n and U_{N+1}^n respectively.

4.7.1. Dirichlet boundary conditions. Let the conservation law (4.1) be augmented with Dirichlet boundary conditions

$$U(x_L, t) = g_L(t), \quad U(x_R, t) = g_R(t). \tag{4.70}$$

These boundary conditions are implemented by specifying the ghost values

$$U_0^n = g_L(t^n), \quad U_{N+1}^n = g_R(t^n). \tag{4.71}$$

Observe that the boundary condition is implemented weakly.

4.7.2. Periodic boundary conditions. These boundary conditions are implemented as

$$U_0^n = U_N^n, \quad U_{N+1}^n = U_1^n. \tag{4.72}$$

4.7.3. Artificial boundary conditions. Non-reflecting Neumann type boundary conditions are implemented as

$$U_0^n = U_1^n, \quad U_{N+1}^n = U_N^n. \tag{4.73}$$

CHAPTER 5

Second-order (high-resolution) finite volume schemes

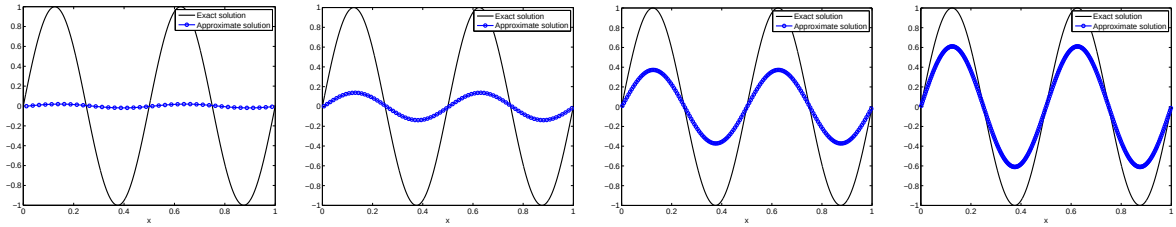
The finite volume schemes (4.14) with suitable numerical fluxes like the Godunov, Lax–Friedrichs and Engquist–Osher fluxes have been demonstrated to be quite robust in approximating scalar conservation laws

$$U_t + f(U)_x = 0. \tag{5.1}$$

However, some problems still persist as the schemes may lead to large errors. To illustrate this point, we consider the linear advection equation (2.2) and compute the solutions with the Godunov (upwind) scheme (2.16) for initial data

$$U(x,0) = \sin(4\pi x) \tag{5.2}$$

in the computational domain $[0,1]$ with periodic boundary conditions.



(a) $N = 40$ (b) $N = 80$ (c) $N = 160$ (d) $N = 320$

FIGURE 5.1. Linear advection equation (2.2) computed to time $t = 10$, using upwind flux with smooth initial conditions (5.2) and periodic boundary conditions at different meshes. `[linAdv_Upw_refine.m]`

No. of Cells	$\mathcal{E}^{\Delta x}$	EOC
20	100	–
40	98.1	0.027
80	86.1	0.188
160	62.7	0.457
320	39	0.688
640	21.9	0.833
1280	11.6	0.914
2560	5.98	0.956

TABLE 5.1. Error and order of convergence for the linear advection equation (2.2) using upwind scheme, with smooth initial data (5.2). `[linAdv_OOC.m]`

The exact solution of the problem (2.2), (5.2) is $U(x,t) = \sin(4\pi(x-t))$. Coupled with the periodic boundary condition, it implies that the exact solution returns back to the initial condition (5.2) at time $t = k$ for all positive integers k. The computed results on a sequence of meshes are presented in Figure

5.1 and Table 5.1. Table 5.1 shows the relative error in L^1 on a sequence of meshes. The percentage relative error is defined by

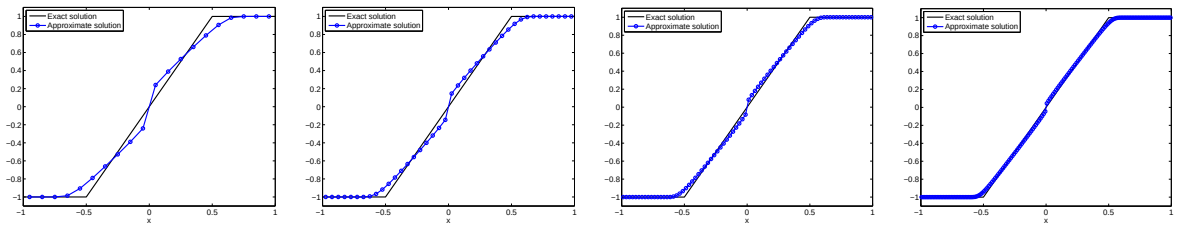
$$\mathcal{E}^{\Delta x} = 100 \times \frac{\|U^{\Delta x} - U^{\text{ref}}\|_{L^1}}{\|U^{\text{ref}}\|_{L^1}}, \tag{5.3}$$

where $U^{\Delta x}$ is the approximate solution computed on a mesh with mesh size Δx and U^{ref} is a reference (exact) solution of the continuous problem. In Table 5.1, we have also shown the *experimental order of convergence* (EOC),

$$\text{EOC}_{\Delta x, \Delta y} = \frac{\log\big(\mathcal{E}^{\Delta x}\big) - \log\big(\mathcal{E}^{\Delta y}\big)}{\log(\Delta x) - \log(\Delta y)}. \tag{5.4}$$

Here, $\Delta x, \Delta y$ are two different mesh sizes. In Table 5.1, we have used $\Delta y = 2\Delta x$ for all the results.

The results of Figure 5.1 indicate that the approximation by the Godunov scheme is quite stable and there are no spurious oscillations or other numerical artifacts. However, the approximation has large errors and the extrema of the solution are clipped. Table 5.1 provides quantitative evidence of these conclusions. The relative errors are quite large, particularly at coarse meshes. An explanation for the large errors is provided in the experimental order of convergence (EOC) column of the table. The observed order of convergence is close to one. This implies that the convergence is slow and the errors are reduced very slowly, impacting the computational efficiency of the scheme.



(a) $N = 20$ (b) $N = 40$ (c) $N = 80$ (d) $N = 160$

FIGURE 5.2. Burgers' equation (3.3) computed up to time $t = 0.5$, using Godunov flux with initial conditions (4.17) and outflow boundary conditions at different meshes. `[burgers_refine.m]`

No. of Cells	$\mathcal{E}^{\Delta x}$	EOC
20	6.86	–
40	4.51	0.605
80	2.84	0.667
160	1.74	0.706
320	1.04	0.745
640	0.606	0.778
1280	0.347	0.804
2560	0.196	0.826

TABLE 5.2. Error and order of convergence for Burgers' equation (3.3) using Godunov's scheme with initial data (4.18). `[burgers_OOC.m]`

A similar situation is observed with computations for the nonlinear Burgers equation (3.3). We compute approximate solutions for initial data (4.18) with the Godunov scheme (4.16) in the computational domain $[-1, 1]$ on a sequence of meshes and present the results in Figure 5.2 and Table 5.2. The results show that the Godunov scheme is stable and converges, but slowly. In fact, the order of convergence is less than one.

Both numerical experiments indicate that the Godunov type finite volume scheme (4.14) is stable but can lead to large errors, due to its slow convergence. A possible recipe for reducing errors is to increase the rate of convergence. This chapter is devoted to introducing the concepts of truncation errors and order of accuracy of schemes. Furthermore, we will construct second-order accurate versions of the finite volume scheme (4.14) for approximating scalar conservation laws.

5.1. Order of accuracy

The order of accuracy of numerical schemes is a very useful concept in numerical analysis.

Definition 5.1. *Assume that a finite volume scheme for approximating* (5.1) *can be written in the generic* $(2p+1)$*-point update form* (4.39)*, and that* $\frac{\Delta t}{\Delta x} \equiv \lambda$ *for some constant* $\lambda > 0$*. The* truncation error *of the scheme is defined as*

$$\mathcal{T}_j^n = U(x_j, t^{n+1}) - H\big(U(x_{j-p}, t^n), \dots, U(x_{j+p}, t^n)\big), \tag{5.5}$$

where U *is the exact solution. The scheme is* q-th order accurate *if* $q \in \mathbb{N}$ *is the largest integer for which*

$$\mathcal{T}_j^n = \mathcal{O}\left(\Delta t^{q+1}\right) \quad \text{for all } j, n \tag{5.6}$$

as $\Delta t \to 0$*.*

Lemma 5.2. *Assume that the exact solution* $U(x,t)$ *of the scalar conservation law is* C^2*, and that a consistent and conservative three-point finite volume method has a* C^2 *update function* H*. Then the scheme is at least first-order accurate.*

Proof. As the solution and the update function H in (4.46) are at least twice continuously differentiable, we can use Taylor expansions. We write the update function as $H = H(X, Y, Z)$. From (4.46) and (4.14) we get

$$\begin{aligned} \frac{\partial H}{\partial X}(U_{j-1}^n, U_j^n, U_{j+1}^n) &= \frac{\Delta t}{\Delta x}\frac{\partial F}{\partial a}(U_{j-1}^n, U_j^n), \\ \frac{\partial H}{\partial Z}(U_{j-1}^n, U_j^n, U_{j+1}^n) &= -\frac{\Delta t}{\Delta x}\frac{\partial F}{\partial b}(U_j^n, U_{j+1}^n). \end{aligned} \tag{5.7}$$

Denoting $V_j^n = U(x_j, t^n)$ and writing down the Taylor expansion for the truncation error (5.5), we obtain

$$\begin{aligned} \mathcal{T}_j^n &= V_j^{n+1} - H\big(V_{j-1}^n, V_j^n, V_{j+1}^n\big) \\ &= V_j^n + \Delta t U_t(x_j, t^n) + \mathcal{O}(\Delta t^2) - H\big(V_j^n, V_j^n, V_j^n\big) \\ &\quad - \frac{\partial H}{\partial X}\big(V_j^n, V_j^n, V_j^n\big)\big(V_{j-1}^n - V_j^n\big) - \frac{\partial H}{\partial Z}\big(V_j^n, V_j^n, V_j^n\big)\big(V_{j+1}^n - V_j^n\big) + \mathcal{O}\left(\Delta x^2\right) \\ &= \Delta t U_t(x_j, t^n) + \mathcal{O}(\Delta t^2) \\ &\quad - \frac{\partial H}{\partial X}\big(V_j^n, V_j^n, V_j^n\big)\big(V_{j-1}^n - V_j^n\big) - \frac{\partial H}{\partial Z}\big(V_j^n, V_j^n, V_j^n\big)\big(V_{j+1}^n - V_j^n\big) + \mathcal{O}\left(\Delta x^2\right) \end{aligned}$$

by the consistency of H. By using the fact that $\Delta t = \lambda \Delta x$ and Taylor expanding once more, the above relation reduces to

$$\begin{aligned} \mathcal{T}_j^n &= \Delta t U_t(x_j, t^n) \\ &\quad + \Delta x \frac{\partial H}{\partial X}\big(V_j^n, V_j^n, V_j^n\big) U_x(x_j, t^n) - \Delta x \frac{\partial H}{\partial Z}\big(V_j^n, V_j^n, V_j^n\big) U_x(x_j, t^n) + \mathcal{O}\left(\Delta t^2\right). \end{aligned}$$

Using the partial derivatives (5.7) and differentiating the consistency relation $F(U,U) = f(U)$ we obtain

$$\begin{aligned} \mathcal{T}_j^n &= \Delta t U_t(x_j, t^n) + \Delta t \left(\frac{\partial F}{\partial a}\big(V_j^n, V_j^n\big) + \frac{\partial F}{\partial b}\big(V_j^n, V_j^n\big) \right) U_x(x_j, t^n) + \mathcal{O}(\Delta t^2) \\ &= \underbrace{\Delta t\big(U_t(x_j, t^n) + f'(U(x_j, t^n)) U_x(x_j, t^n)\big)}_{=\,0 \text{ by } (5.1)} + \mathcal{O}\left(\Delta t^2\right) \\ &= \mathcal{O}\left(\Delta t^2\right). \end{aligned}$$

Thus (5.6) holds for $q = 1$. □

Note that the conditions of Lemma 5.2 are satisfied by the Lax–Friedrichs and Engquist–Osher schemes. However, the Godunov flux (4.15) is not C^2 and the conclusions of the above lemma do not directly apply to the Godunov flux, except in the linear case. Nevertheless, it may be shown that the Godunov flux is formally first-order accurate.

The fact that the truncation error has a certain decay as $\Delta t \to 0$ can be converted to an estimate on the rate of convergence in the linear case. To be more precise, consider the transport equation (2.2) and a first-order scheme (first-order in the sense of (5.6) with $q = 1$). Denote the computed solutions for a mesh size Δx as $U^{\Delta x}$, and the exact solution as U, both evaluated at some time t. Then it may be shown (see e.g. [**TW09**]) that

$$\|U^{\Delta x} - U\|_{L^1(\mathbb{R})} \leqslant C\Delta x.$$

Thus a first-order scheme for a linear equation has a unit rate of convergence to the exact solution. This rate of convergence is demonstrated in the numerical experiments for the linear equation (2.2), presented earlier in this chapter.

However, a similar result does not hold for the nonlinear conservation law (5.1). The truncation error is only defined for smooth solutions, as the Taylor expansion is needed to obtain a truncation error estimate. It is well established by now that solutions of (5.1) are discontinuous and that the Taylor expansion is not valid for such solutions. Hence, the notion of truncation error is *purely formal* for the nonlinear case. A famous result of Kuznetsov [**GR91**] shows that monotone schemes (see Chapter 4 for definitions) satisfy an estimate of the form

$$\|U^{\Delta x} - U\|_{L^1(\mathbb{R})} \leqslant C\Delta x^{1/2}.$$

Hence, formally first-order schemes like the Godunov, Lax–Friedrichs and related schemes may show a rate of convergence lower than unity. However, the rate of convergence is close to one for most numerical experiments as the ones presented at the beginning of this chapter.

5.1.1. The Lax–Wendroff scheme. In order to obtain better resolution of the approximate solutions of (5.1), we need to design numerical schemes that are better than first-order accurate, at least formally. The simplest recipe for obtaining a higher order scheme is to use Taylor expansions more effectively. Let U be a solution of (5.1) and assume that it is smooth. We then have

$$\begin{aligned} U_t &= -f(U)_x, \\ U_{tt} &= -(f(U)_x)_t = -(f(U)_t)_x = -(f'(U)U_t)_x = (f'(U)f(U)_x)_x. \end{aligned} \tag{5.8}$$

Expanding the exact solution in terms of Taylor expansions and using the identities (5.8), we obtain

$$\begin{aligned} U(x_j, t^{n+1}) &= U(x_j, t^n) + \Delta t U_t(x_j, t^n) + \frac{\Delta t^2}{2} U_{tt}(x_j, t^n) + \mathcal{O}\left(\Delta t^3\right) \\ &= U(x_j, t^n) - \Delta t f(U(x_j, t^n)_x \\ &\quad + \frac{\Delta t^2}{2}\left(f'(U(x_j, t^n))f(U(x_j, t^n))_x\right)_x + \mathcal{O}\left(\Delta t^3\right). \end{aligned}$$

We approximate the spatial derivatives with second-order accurate central differences as

$$\begin{aligned} f(U)_x &\approx \frac{f(U^n_{j+1}) - f(U^n_{j-1})}{2\Delta x}, \\ (f'(U)f(U)_x)_x &\approx \frac{1}{\Delta x}\left(a^n_{j+1/2}\left(\frac{f(U^n_{j+1}) - f(U^n_j)}{\Delta x}\right) - a^n_{j-1/2}\left(\frac{f(U^n_j) - f(U^n_{j-1})}{\Delta x}\right)\right), \end{aligned}$$

and obtain the *Lax–Wendroff scheme*

$$\begin{aligned} U^{n+1}_j = U^n_j &- \frac{\Delta t}{2\Delta x}\left(f(U^n_{j+1}) - f(U^n_{j-1})\right) \\ &+ \frac{\Delta t^2}{2\Delta x^2}\left(a^n_{j+1/2}(f(U^n_{j+1}) - f(U^n_j)) - a^n_{j-1/2}(f(U^n_j) - f(U^n_{j-1}))\right). \end{aligned} \tag{5.9}$$

Here,

$$a^n_{j+1/2} = f'\left(\frac{U^n_j + U^n_{j+1}}{2}\right)$$

is an approximation to $f'(U(x_{j+1/2},t))$. The Lax–Wendroff scheme can be written in the standard finite volume form (4.14) with the numerical flux function

$$F^n_{j+1/2} = F(U^n_j, U^n_{j+1}) = \frac{f(U^n_j)+f(U^n_{j+1})}{2} - \frac{a^n_{j+1/2}\Delta t}{2\Delta x}\big(f(U^n_{j+1})-f(U^n_j)\big). \tag{5.10}$$

Exercise 5.3. *Show that the Lax–Wendroff scheme is second-order accurate in the sense of* (5.6).

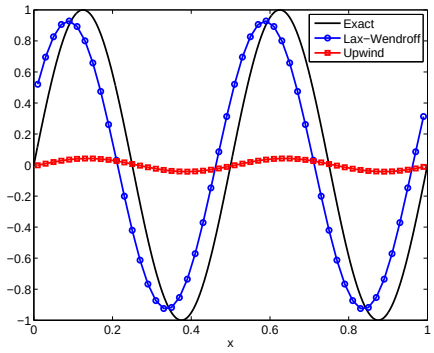


FIGURE 5.3. Linear advection equation computed to time $t = 10$ on 50 mesh points, using the Lax–Wendroff and upwind schemes with smooth initial conditions (5.2) and periodic boundary conditions. `[linAdv_LW_Upw_sine.m]`

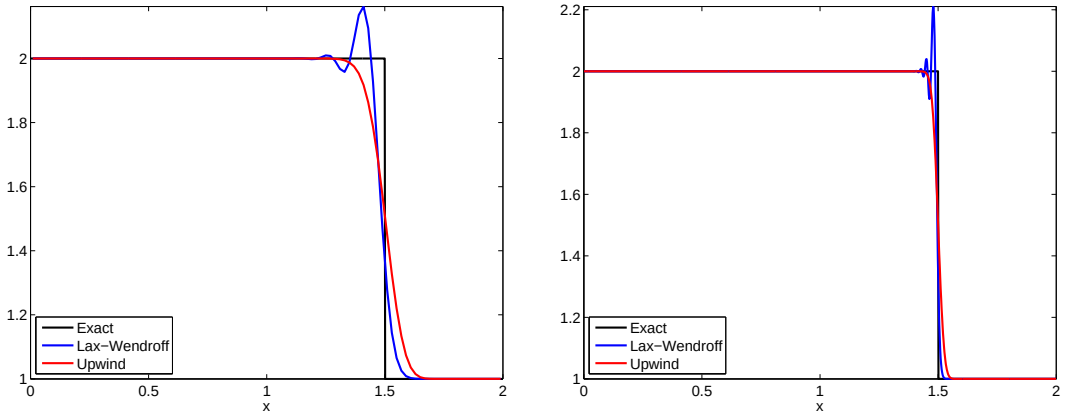
No. of Cells	$\mathcal{E}^{\Delta x}$	EOC
20	132	–
40	67.3	0.967
80	18.4	1.87
160	4.64	1.98
320	1.16	2
640	0.291	2
1280	0.0727	2
2560	0.0182	2

TABLE 5.3. Error and order of convergence for the linear advection equation using the Lax–Wendroff scheme, with smooth initial data (5.2). `[linAdv_LW_OOC.m]`

5.1.2. Numerical experiments. We perform several numerical experiments with the Lax–Wendroff scheme. To begin with, we consider the advection equation (2.2) with initial data (5.2) and periodic boundary conditions. The numerical results are presented in Figure 5.3. For the sake of comparison, we plot the results obtained with the Godunov (upwind) scheme. The Lax–Wendroff scheme is clearly more accurate. Quantitative results in error Table 5.3 attest to the fact that Lax–Wendroff has a rate of convergence equal to 2, thus justifying its derivation as a second-order scheme.

In the next numerical experiment, the Lax–Wendroff scheme is used to compute solutions of the linear advection equation (2.2) with Riemann data (2.27). The results in Figure 5.4(a) show that the discontinuity is resolved much more sharply by the Lax–Wendroff scheme when compared to the Godunov scheme. However, the Lax–Wendroff scheme produces incorrect oscillations in the wake of the shock. These oscillations increase in both amplitude and frequency as the mesh is refined (see Figure 5.4(b)). Thus, increasing the order of accuracy may lead to stability issues, at least with the Lax–Wendroff scheme.

As another numerical example, consider the nonlinear Burgers equation (3.3) with Riemann data (4.17). The results, shown in Figure 5.5, show that although the Lax–Wendroff scheme resolves the shock slightly better than the Godunov scheme, the solution is polluted with spurious oscillations. In conclusion, the Lax–Wendroff scheme fails to resolve discontinuous solutions of conservation laws in a stable, monotonous manner.



(a) 100 grid points (b) 1000 grid points

FIGURE 5.4. Linear advection equation computed to time $t = 1$ with discontinuous initial conditions (2.27) and outflow boundary conditions. Comparison of the Lax–Wendroff and upwind schemes. `[linAdv_LW_Upw_disc.m]`

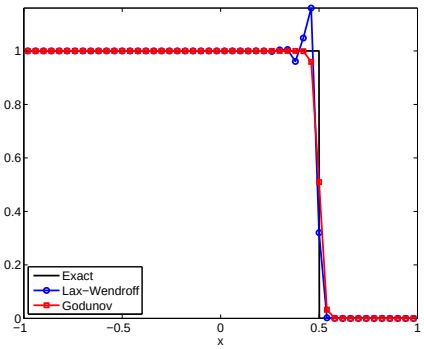


FIGURE 5.5. Burgers' equation computed to time $t = 0.5$ with discontinuous initial conditions (4.17) and outflow boundary conditions on 50 cells. `[burgers_LW_God_disc.m]`

5.2. The REA algorithm

Given the instabilities that result from the Lax–Wendroff scheme, we need to devise a new procedure for obtaining high-order schemes. This forces us to revisit the original derivation of the Godunov scheme, presented in Chapter 4. It turns out that the entire derivation of the Godunov scheme can be summarized in the following three steps:

Reconstruction: At time level t^n, assume that we know the approximate cell averages U_j^n. We realize this collection of cell average by a piecewise constant function

$$U(x, t^n) = U_j^n \qquad \text{for } x_{j-1/2} < x < x_{j+1/2}. \tag{5.11}$$

Evolution: The reconstructed function $U(x, t^n)$ is evolved in time using either an exact or approximate solution algorithm for the conservation law. This amount to solving the superposition of Riemann problems (4.8), either exactly or approximately.

Averaging: Last, we average the solution at the next time step t^{n+1} over each control volume $\mathcal{C}$.

The three steps of reconstruction, evolution and averaging are branded together as the REA algorithm. Most of the finite volume schemes of the previous chapter can be obtained from the REA algorithm by a suitable approximation of the evolution step.

The main constraint of the REA algorithm so far is that the reconstruction step utilizes only piecewise constant functions (Figure 5.6). For smooth solutions of (5.1), piecewise constant functions are a first-order interpolation, thus resulting in the overall first-order accuracy of the Godunov-type schemes. One possible recipe for constructing high-order accurate schemes lies in employing high-order interpolations in the reconstruction step.

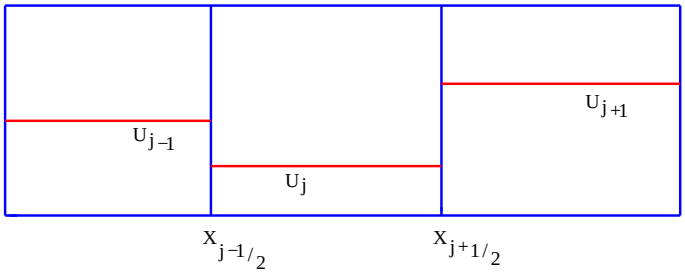


FIGURE 5.6. Representation of cell averages as piecewise constant functions in the REA algorithm.

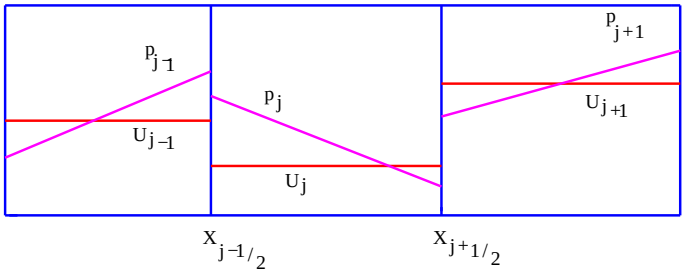


FIGURE 5.7. Piecewise linear functions in the REA algorithm.

5.2.1. Second-order reconstruction. The simplest high-order interpolation results from a piecewise linear reconstruction (Figure 5.7). Such a reconstruction will lead to a second-order accurate approximation of the smooth solutions of (5.1). Given the cell averages U_j^n at time t^n, there are several possibilities for reconstructing linear functions in each cell $\mathcal{C}_j$. We need to narrow down the alternatives by putting constraints on the reconstruction.

One of the key requirements for a stable and convergent approximation of conservation laws is that the scheme is conservative, (4.40).

Exercise 5.4. *Show that the evolution and averaging steps are conservative by verifying that*

(a) if $U(x,t^{n+1})$ is the exact solution at time t^{n+1}, then

$$\int_{\mathbb{R}} U(x,t^{n+1})\,dx = \int_{\mathbb{R}} U(x,t^n)\,dx,$$

(b) if U_j^{n+1} is the average of $U(x,t^{n+1})$ in cell j, then

$$\Delta x \sum_j U_j^{n+1} = \int_{\mathbb{R}} U(x,t^{n+1})\,dx.$$

It is natural to demand that the reconstruction step is also conservative. Denoting the piecewise linear function in the cell $\mathcal{C}_j$ as $p_j(x)$, we require that

$$\frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} p_j(x)\,dx = U_j^n. \tag{5.12}$$

It is readily seen that p_j must have the form

$$p_j(x) = U_j^n + \sigma_j^n(x - x_j), \tag{5.13}$$

where σ_j^n is a parameter that determines the slope in cell $\mathcal{C}_j$. The *local* linear functions p_j are combined to define the *global* piecewise linear function

$$p(x) = p_j(x) \qquad \text{for } x_{j-1/2} < x < x_{j+1/2}. \tag{5.14}$$

This piecewise linear function can be used with the REA algorithm to obtain a higher order scheme.

5.2.2. Choices of slope. The slope σ in (5.13) can be determined in a variety of ways. Three of them are

- Central:

$$\sigma_j^n = \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x}. \tag{5.15}$$

- Backward:

$$\sigma_j^n = \frac{U_j^n - U_{j-1}^n}{\Delta x}. \tag{5.16}$$

- Forward:

$$\sigma_j^n = \frac{U_{j+1}^n - U_j^n}{\Delta x}. \tag{5.17}$$

It is instructive to see what scheme the REA algorithm with the above choices of slope results in. For simplicity of the exposition, we consider the linear advection equation (2.2) with a positive velocity $a > 0$.

Exercise 5.5. *Let* (5.14) *be the prescribed initial data for the linear advection equation* (2.2) *at time* t^n*, and assume that* $a > 0$*. Show that the cell average* U_j^{n+1} *of the exact solution at time* t^{n+1} *can be written as*

$$\begin{aligned} U_j^{n+1} &= \frac{a\Delta t}{\Delta x}\left(U_{j-1}^n + \frac{1}{2}(\Delta x - a\Delta t)\sigma_{j-1}^n\right) + \left(1 - \frac{a\Delta t}{\Delta x}\right)\left(U_j^n - \frac{1}{2}a\Delta t\sigma_j^n\right) \\ &= U_j^n - \frac{a\Delta t}{\Delta x}\big(U_j^n - U_{j-1}^n\big) - \frac{1}{2}\frac{a\Delta t}{\Delta x}(\Delta x - a\Delta t)(\sigma_j^n - \sigma_{j-1}^n). \end{aligned} \tag{5.18}$$

Choosing σ as the downwind slope (5.17), we obtain the following explicit formula for (5.18):

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}(U_{j+1}^n - U_{j-1}^n) + \frac{a^2\Delta t^2}{2\Delta x^2}(U_{j+1}^n - 2U_j^n + U_{j-1}^n). \tag{5.19}$$

This is precisely the Lax–Wendroff scheme (5.9) for the linear advection equation. This conclusion also justifies the fact that a second-order piecewise linear reconstruction leads to an overall second-order accurate scheme like the Lax–Wendroff scheme. However, the Lax–Wendroff scheme produces oscillations, and so the choice of a downwind slope is not desirable.

Next, we consider the upwind slope (5.16). The explicit formula (5.18) then becomes

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}(3U_j^n - 4U_{j-1}^n + U_{j-2}^n) + \frac{a^2\Delta t^2}{2\Delta x^2}(U_j^n - 2U_{j-1}^n + U_{j-2}^n). \tag{5.20}$$

This four point scheme is called the Beam–Warming scheme. An approximate solution of the linear advection equation with Riemann data (2.27) computed with the Beam–Warming scheme is shown in Figure 5.8. The results show that the accuracy of the scheme at the shock is comparable to the Lax–Wendroff scheme. However, it also leads to oscillations. Similarly, using the central slope (5.15) leads to oscillatory approximations.

5.2.3. The source of oscillations. The exact solution for the linear transport equation (2.2) as well as the nonlinear conservation law (5.1) are TVD and consequently nonoscillatory whenever the initial data is nonoscillatory. It is reasonable to demand that stable numerical schemes respect this property (see Section 4.5). The numerical results presented above demonstrate that second-order schemes like the Lax–Wendroff (5.9) and the Beam–Warming schemes (5.20) violate the requirement that the approximate solutions be TVD. What goes wrong and why do these schemes violate this requirement?

A closer look at the REA algorithm reveals that the evolution step is TVD whenever an exact solution of the Riemann problem is used. Furthermore, suitable approximate Riemann solvers like the

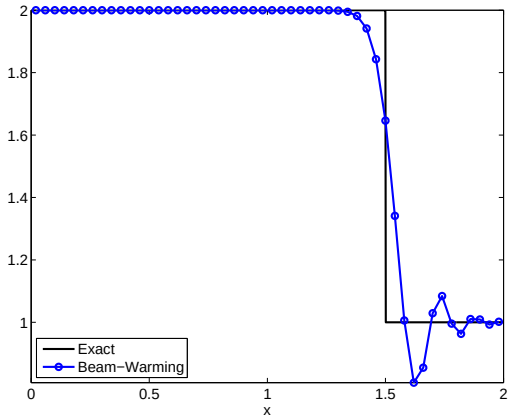


FIGURE 5.8. Linear advection equation computed to time $t = 1$ with discontinuous initial conditions (2.27) and outflow boundary conditions on 50 cells using the Beam–Warming scheme. `linAdv_beamWarm_disc.m`]

Lax–Friedrichs solver are also TVD. Finally, it is straightforward to check that the averaging operator respects the TVD property. Indeed, considering $u \in BV(\mathbb{R})$ and u_Δ defined by:

$$u_\Delta(x) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(y)\, dy,$$

we see that

$$\begin{aligned} TV(u_\Delta) &= \sum_{j\in\mathbb{Z}} \frac{1}{\Delta x} \left| \int_{x_{j+1/2}}^{x_{j+3/2}} u(y)\, dy - \int_{x_{j-1/2}}^{x_{j+1/2}} u(y)\, dy \right| \\ &= \sum_{j\in\mathbb{Z}} \frac{1}{\Delta x} \left| \int_0^{\Delta x} u(x_{j-1/2} + \Delta x + s) - u(x_{j-1/2} + s) ds \right| \\ &\leqslant \frac{1}{\Delta x} \int_0^{\Delta x} \sum_{j\in\mathbb{Z}} \left| u(x_{j-1/2} + \Delta x + s) - u(x_{j-1/2} + s) \right| ds \\ &\leqslant \frac{1}{\Delta x} \int_0^{\Delta x} \|u(\cdot + s)\|_{TV(\mathbb{R})}\, ds = TV(u). \end{aligned}$$

The only explanation for the fact that the Lax–Wendroff and Beam–Warming schemes are not TVD is that the second-order reconstruction (5.13) violates the TVD requirement. To illustrate this point, we consider the following simple situation: Let $J \in \mathbb{Z}$ be fixed and consider cell averages at time level t^n

$$U_j^n = \begin{cases} 1 & \text{if } j \leqslant J \\ 0 & \text{if } j > J. \end{cases} \tag{5.21}$$

The initial total variation is 1. Computing the downwind slope (5.17), we find that $\sigma_j^n = 0$ for $j \neq J$ and $\sigma_J^n = -0.5$. Therefore, the piecewise linear reconstruction (5.17) is active only in the cell J (see Figure 5.9). Furthermore, the reconstructed function has an overshoot with a maximum value of 1.5. The resulting total variation is 1.5, which is greater than the total variation of the initial cell averages. This induction of oscillations is independent of Δx and can lead to much larger oscillations as the solutions are evolved in time. Similar results hold for the upwind and central slopes.

Given the nature of the continuous solution, it is reasonable to expect that the reconstruction (5.13) is TVD, i.e,

$$\|p\|_{BV} \leqslant \|U^{\Delta x}\|_{BV}, \tag{5.22}$$

where p is the piecewise linear function (5.14) and $U^{\Delta x}$ is the piecewise constant function (5.11).

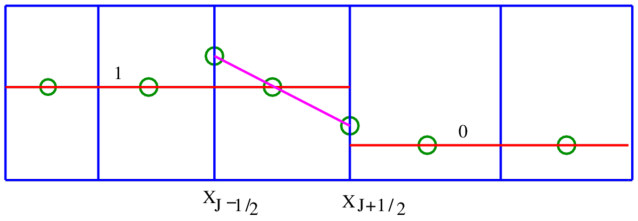


FIGURE 5.9. The Lax–Wendroff reconstruction for (5.21).

5.3. The minmod limiter

Straightforward choices of slope like the central (5.15), upwind (5.16) and downwind (5.17) slopes do not satisfy the TVD requirement (5.22). The problem is manifested near discontinuities as shown in Figure 5.9.

A clever choice of the slope in (5.13) that satisfies the TVD property (5.22) is the so-called *minmod* limiter, which is given by

$$\sigma_j^n = \operatorname{minmod}\left(\frac{U_{j+1}^n - U_j^n}{\Delta x},\ \frac{U_j^n - U_{j-1}^n}{\Delta x}\right). \tag{5.23}$$

The minmod function is defined as

$$\operatorname{minmod}(a_1,\dots,a_n) = \begin{cases} \operatorname{sign}(a_1)\min\limits_{1\leqslant k\leqslant n}(|a_k|) & \text{if } \operatorname{sign}(a_1) = \cdots = \operatorname{sign}(a_n), \\ 0 & \text{otherwise.} \end{cases} \tag{5.24}$$

The minmod limiter compares the upwind slope and the downwind slope and checks if they are of the same sign. If so, it selects the smallest one, and if not, it sets the slope to zero. Thus, the reconstruction is limited based on local gradients.

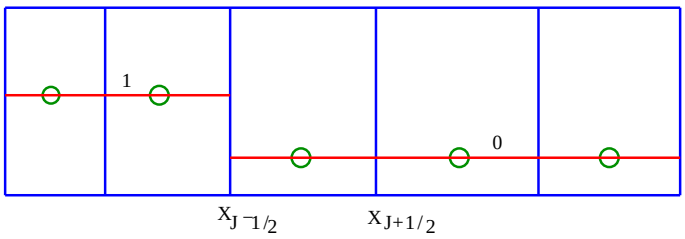


FIGURE 5.10. The minmod reconstruction for (5.21).

To illustrate the effect of the minmod limiter, we take the same example as before and consider initial data (5.21). Using the minmod slope (5.23) in (5.13) results in retaining the initial data (5.21) (see Figure 5.10). Thus, at least in this case, the minmod reconstruction is TVD.

Next, we consider cell averages

$$U_j^n = \begin{cases} 1 & \text{if } j \leqslant J-1 \\ 3/4 & \text{if } j = J \\ 1/4 & \text{if } j = J+1 \\ 0 & \text{if } j \geqslant J+2. \end{cases} \tag{5.25}$$

The reconstructed piecewise linear function (5.13) with minmod slopes (5.23) is depicted in Figure 5.11. The reconstruction is clearly TVD.

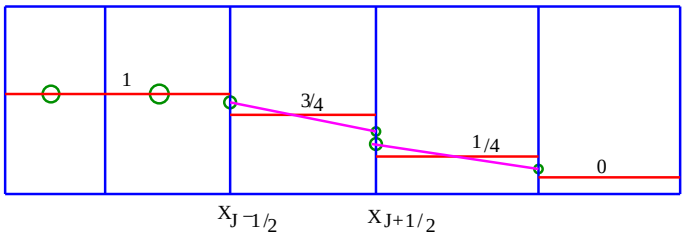
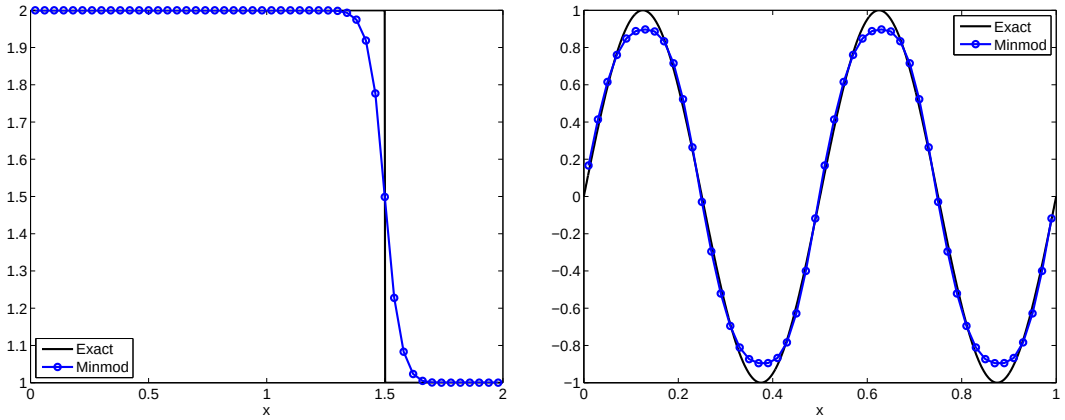


FIGURE 5.11. The minmod reconstruction for (5.25).

Exercise 5.6. *Show that the minmod reconstruction $p(x)$ of cell averages U_j is TVD: $\|p\|_{\mathrm{TV}} \leqslant \sum_j |U_{j+1} - U_j|$.*



(a) Discontinuous initial data (2.27)

(b) Smooth initial data (5.2)

FIGURE 5.12. Linear advection equation computed to time $t = 1$ 50 cells using the minmod limiter. `[linAdv_reconstr.m]`

No. of cells	Minmod		MC		Superbee	
	Relative error	EOC	Relative error	EOC	Relative error	EOC
20	94.6	–	68.4	–	49.1	–
40	54.4	0.798	14.7	2.22	12.4	1.98
80	17	1.68	4.93	1.57	3.97	1.64
160	7.7	1.14	1.51	1.71	3.64	0.127
320	2.14	1.85	0.407	1.89	1.44	1.34
640	0.613	1.8	0.104	1.97	0.416	1.79
1280	0.168	1.87	0.0262	1.99	0.112	1.89
2560	0.0449	1.9	0.00657	1.99	0.0292	1.94

TABLE 5.4. Error and order of convergence for the linear advection equation with smooth initial data (5.2). `[linAdv_reconstr_OOC.m]`

5.3.1. Numerical experiments. We test the linear advection equation (2.2) with the second-order scheme (5.18) and a minmod slope limiter (5.23). To begin with, we use the initial data (2.27) and see that the resulting scheme is not oscillatory (see Figure 5.12(a)). The scheme improves the accuracy considerably when compared to the first-order scheme, and captures the discontinuity quite sharply.

Next, we test the minmod scheme with smooth initial data (5.2) and show the resulting profiles in Figure 5.12(b). The minmod scheme resolves the solution quite well. Error table 5.4 shows that

the scheme is close to second-order accurate. Numerical experiments with nonlinear examples will be presented in the sequel.

5.4. Other limiters

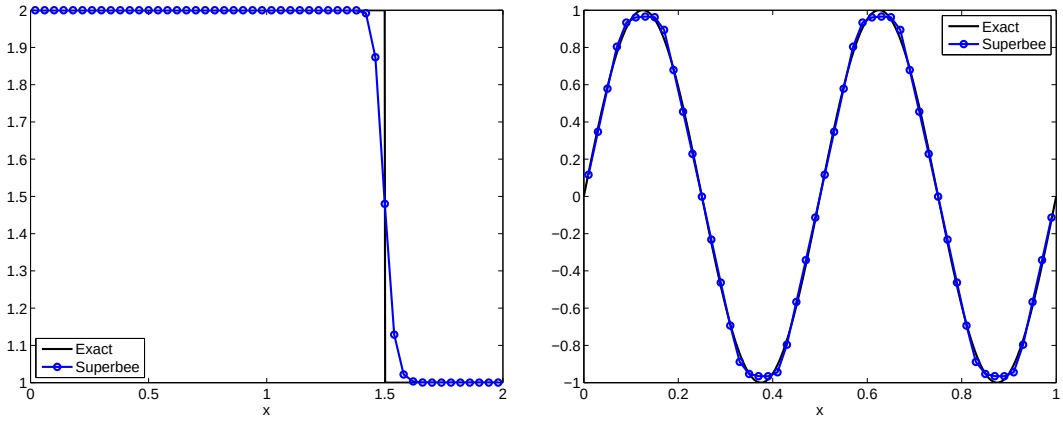
We have demonstrated that the minmod limiter (5.23) improves the quality of the solution considerably. However, other choices of slope limiters exist and might result in better approximations of the underlying PDE. A popular choice is the *superbee* limiter, due to Roe, which has the expression

$$\sigma_j^n = \text{maxmod}\left(\sigma_j^L, \sigma_j^R\right), \tag{5.26}$$

where

$$\sigma_j^L = \text{minmod}\left(2\frac{U_j^n - U_{j-1}^n}{\Delta x}, \frac{U_{j+1}^n - U_j^n}{\Delta x}\right),$$
$$\sigma_j^R = \text{minmod}\left(\frac{U_j^n - U_{j-1}^n}{\Delta x}, 2\frac{U_{j+1}^n - U_j^n}{\Delta x}\right),$$

with the maxmod function simply replacing the minimum in (5.24) with a maximum.



(a) Discontinuous initial data (2.27)

(b) Smooth initial data (5.2)

FIGURE 5.13. Linear advection equation computed to time $t = 1$ 50 cells using the superbee limiter. `[linAdv_reconstr.m]`

Using the superbee limiter with the initial data (5.25) results in a piecewise linear function shown in Figure 5.14. The figure indicates that the superbee limiter results in a TVD reconstruction. Furthermore, comparing the superbee limiter with the minmod limiter in Figure 5.11 shows that the superbee limiter results in steeper slopes, while still being TVD. Employing the superbee limiter in the scheme (5.18) to compute approximate solutions for the linear advection equation (2.2), we present results for initial data (2.27) and (5.2) in Figure 5.13 (a) and (b), respectively. The results show that the superbee limiter gives accurate and nonoscillatory approximations. The superbee limiter is more accurate than the minmod limiter (particularly at the extrema). However, it has a tendency to square off extrema, as shown in Figure 5.13(b). Error table 5.4 shows that second-order accuracy is obtained.

Another popular limiter is the MC (monotonized central) limiter of Van-Leer,

$$\sigma_j^n = \text{minmod}\left(2\frac{U_{j+1}^n - U_j^n}{\Delta x}, \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta x}, 2\frac{U_j^n - U_{j-1}^n}{\Delta x}\right). \tag{5.27}$$

Numerical results (see Figure 5.15 and Table 5.4) show that the MC limiter is as accurate as the superbee limiter. It is usually the preferred slope limiter. Yet another choice is van Leer's limiter:

$$\sigma_j^n = \frac{r + |r|}{1 + |r|}, \qquad \text{where } r = \frac{U_{j+1}^n - U_j^n}{U_j^n - U_{j-1}^n}.$$

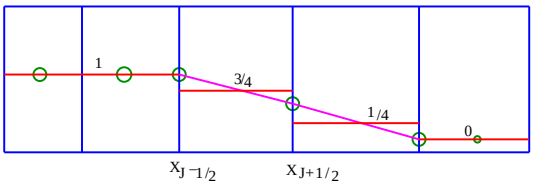


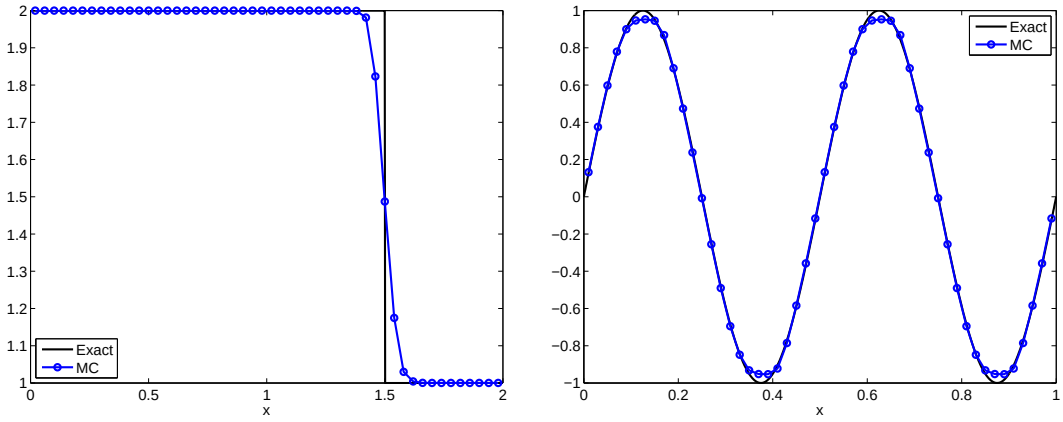
FIGURE 5.14. The superbee reconstruction for (5.25).

5.4.1. Second-order schemes in the flux form. So far, we have written the second-order schemes for the linear advection equation (2.2) in the update form (5.18). They can easily be recast into the finite volume flux form (4.14). Assume for the moment that the advection speed $a > 0$. Recall that the interface flux $F_{i+1/2}$ was calculated as the interface integral (4.13). Computing this integral with the exact solution (2.5) and the second-order reconstruction (5.13), we obtain

$$\begin{aligned}
F^n_{j+1/2} &= \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(U(x_{j+1/2},t)\,dt \\
&= \frac{a}{\Delta t}\int_{t^n}^{t^{n+1}} U(x_{j+1/2},t)\,dt && (\text{as } f(U)=aU) \\
&= \frac{a}{\Delta t}\int_{t^n}^{t^{n+1}} p_j\left(x_{j+1/2}-a(t-t^n)\right)\,dt && (\text{from (2.5)}) \\
&= \frac{a}{\Delta t}\int_{t^n}^{t^{n+1}} U^n_j+\sigma^n_j\left(x_{j+1/2}-a(t-t^n)-x_j\right)\,dt && (\text{from (5.13)}) \\
&= aU^n_j+\frac{a}{2\Delta t}\int_{t^n}^{t^{n+1}} \sigma^n_j\big(\Delta x-2a(t-t^n)\big)\,dt \\
&= aU^n_j+\frac{a}{2}(\Delta x-a\Delta t)\sigma^n_j.
\end{aligned}$$

Consequently, the second-order scheme (5.18) can be written in the finite volume form (4.14) with flux

$$F^n_{j+1/2} = aU^n_j+\frac{a}{2}(\Delta x-a\Delta t)\sigma^n_j. \tag{5.28}$$



(a) Discontinuous initial data (2.27)

(b) Smooth initial data (5.2)

FIGURE 5.15. Linear advection equation computed to time $t = 1$ 50 cells using the MC limiter. `[linAdv_reconstr.m]`

A similar expression can be derived when the advection velocity $a < 0$. Defining $\delta^n_{j+1/2} = \Delta x \sigma^n_j$, we can rewrite (5.28) as

$$F^n_{j+1/2} = aU^n_j + \frac{a}{2}\left(1 - \frac{a\Delta t}{\Delta x}\right)\delta^n_{j+1/2}. \tag{5.29}$$

Denote the jump of the solution at the interface $x_{j+1/2}$ as

$$[\![U^n]\!]_{j+1/2} = U^n_{j+1} - U^n_j$$

and define

$$\theta^n_{j+1/2} = \frac{[\![U^n]\!]_{j-1/2}}{[\![U^n]\!]_{j+1/2}}.$$

The parameter δ is an indication of the change in U around $x_{j+1/2}$. Therefore, we rewrite it in terms of cell interface jumps,

$$\delta^n_{j+1/2} = \varphi(\theta^n_{j+1/2})[\![U^n]\!]_{j+1/2}$$

for some function $\varphi(\theta)$. The advantage of writing δ in this *limited* form is immediate. If $\varphi \equiv 1$, then the flux (5.29) immediately reduces to the Lax–Wendroff flux (5.9) for the linear advection equation. Hence, the second-order scheme can be thought as a version of the Lax–Wendroff scheme with the flux being limited by the limiter function φ. In conclusion, a limiter on the slope of the linear function (5.13) can be written as a *flux limiter*.

5.5. Flux limiters and the TVD property.

We can realize slope limiters as flux limiters. Table 5.5 gives the flux limiter form of the slope limiters we have studied so far.

Exercise 5.7. *For each flux limiter in Table 5.5, show that the resulting scheme is the same as when using the corresponding slope limiter in* (5.18).

Method	Flux limiter function $\varphi(\theta)$
Upwind	0
Lax–Wendroff	1
Beam–Warming	θ
Minmod	$\operatorname{minmod}(1,\theta)$
Superbee	$\max(0, \min(1, 2\theta), \min(2, \theta))$
MC	$\max\left(0, \min\left(\frac{1+\theta}{2}, 2, 2\theta\right)\right)$
van Leer	$\frac{\theta+\lvert\theta\rvert}{1+\lvert\theta\rvert}$

TABLE 5.5. The most popular flux limiters.

Observe that the upwind, Lax–Wendroff and Beam–Warming flux limiters are linear (or linear affine), whereas the minmod, superbee and MC limiters are nonlinear. It is essential to consider nonlinear limiters to obtain TVD second-order accurate schemes, as we will see in a moment.

We can rewrite the second-order scheme (5.18) in the incremental form

$$U^{n+1}_j = U^n_j + C^n_{j+1/2}[\![U^n]\!]_{j+1/2} - D^n_{j-1/2}[\![U^n]\!]_{j-1/2} \tag{5.30}$$

(consult (4.55)) with coefficients

$$C^n_{j+1/2} = -\frac{a}{2}\lambda(1 - a\lambda)\varphi\left(\theta^n_{j+1/2}\right),$$
$$D^n_{j-1/2} = a\lambda - \frac{a}{2}\lambda(1 - a\lambda)\varphi\left(\theta^n_{j-1/2}\right),$$

where $\lambda = \frac{a\Delta t}{\Delta x}$. However, realizing that $[\![U^n]\!]_{j+1/2} = \frac{[\![U^n]\!]_{j-1/2}}{\theta^n_{j+1/2}}$, we rewrite the coefficients in (5.30) in the more revealing form

$$
\begin{aligned}
C^n_{j+1/2} &= 0, \\
D^n_{j-1/2} &= a\lambda + \frac{a}{2}\lambda(1-a\lambda)\left(\frac{\varphi(\theta^n_{j+1/2})}{\theta^n_{j+1/2}} - \varphi(\theta^n_{j-1/2})\right).
\end{aligned} \tag{5.31}
$$

By Harten's Lemma 4.12, the scheme will be TVD if the coefficients C and D in (5.31) satisfy the criteria (4.57). Hence, we have to ensure that

$$0 \leqslant D^n_{j-1/2} \leqslant 1. \tag{5.32}$$

We recall that $\lambda \leqslant 1$ due to the CFL condition (2.18). Hence, a sufficient condition for (5.32) to hold is given by

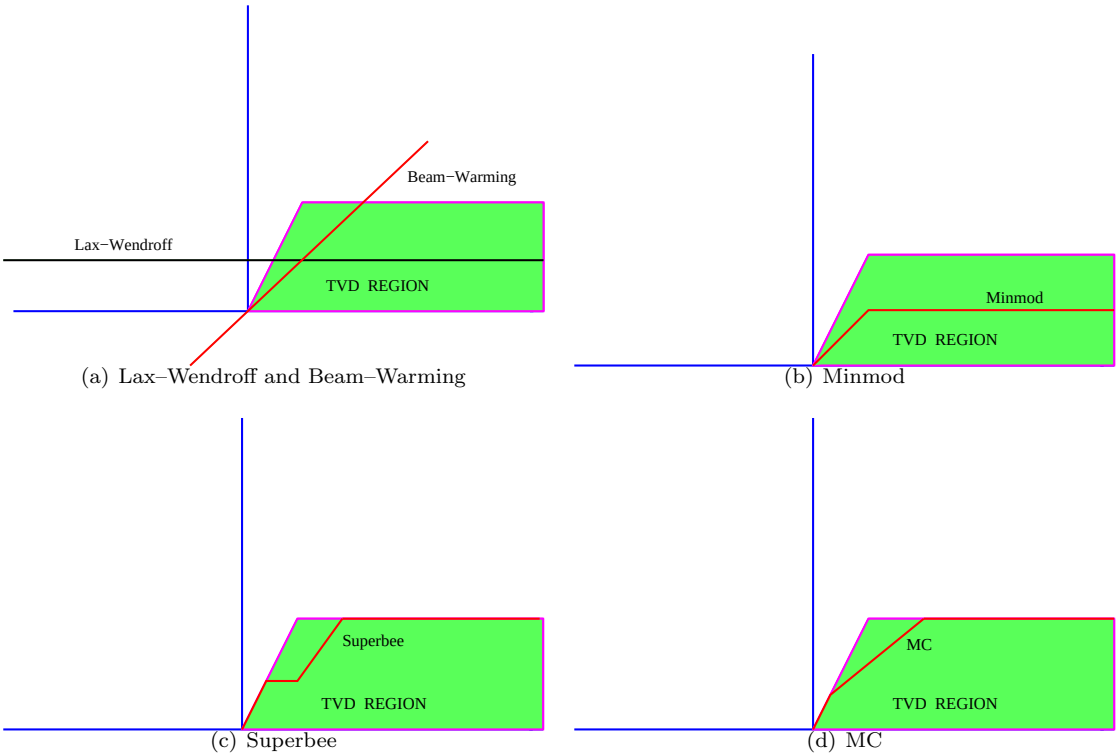
$$\left|\frac{\varphi(\theta_1)}{\theta_1} - \varphi(\theta_2)\right| \leqslant 2 \quad \text{for } \theta_1, \theta_2 \in \mathbb{R}. \tag{5.33}$$

The set of solutions to (5.33) is rich. A particular class of solutions is given by

$$0 \leqslant \varphi(\theta) \leqslant \operatorname{minmod}(2, 2\theta) \quad \text{for } \theta \in \mathbb{R}. \tag{5.34}$$

It is straightforward to verify that (5.34) satisfies (5.33). The region defined by (5.34) is shown in Figure 5.16. Note that (5.34) requires that $\varphi \equiv 0$ whenever $\theta < 0$. This occurs precisely when there is a local extrema at x_j.

Exercise 5.8. *Show that any flux limiter that satisfies* (5.34) *reduces to first-order accuracy at extrema. In particular, show that when U^n_j has a local maximum or minimum in cell j, then $p_j(x) \equiv U^n_j$.*



(a) Lax–Wendroff and Beam–Warming (b) Minmod

(c) Superbee (d) MC

FIGURE 5.16. Sweby diagrams for different limiters.

Exercise 5.9. *Verify that the Lax–Wendroff method and the Beam–Warming method do not satisfy the TVD requirement* (5.34)*, whereas the minmod, superbee and MC limiters do satisfy* (5.34).

5.6. High-resolution methods for nonlinear problems.

The limiter-based methods described above can be directly used for approximating the nonlinear equation (5.1), as the REA algorithm applies to any conservation law. However, it may be difficult to obtain explicit formulas like (5.28) in this case, as the flux integral (4.13) may not be possible to evaluate explicitly. Furthermore, an analogue of the exact solution (2.5) is not available in the nonlinear case, making an evaluation of (4.13) is very complicated. Therefore, we use a related but slightly different approach to yield high-resolution methods for nonlinear equations.

5.6.1. Semi-discrete formulation. The starting point for obtaining high-resolution methods for the nonlinear conservation law (5.1) is still the reconstruction procedure, as described previously. Instead of employing the REA algorithm directly, we denote the cell average over $\mathcal{C}_j$ as

$$U_j(t) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} U(x,t)\, dx$$

and integrate the conservation law (5.1) over space to obtain

$$\frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} U_t + f(U)_x \, dx = 0$$

$$\Rightarrow \qquad \frac{d}{dt} U_j(t) + \frac{1}{\Delta x} \Big(f(U(x^-_{j+1/2}, t)) - f(U(x^+_{j-1/2}, t)) \Big) = 0.$$

Here, $U(x^-_{j+1/2}, t)$ denotes the left-limit of U at $x_{j+1/2}$,

$$U(x^-_{j+1/2}, t) = \lim_{x \to x^-_{j+1/2}} U(x,t).$$

Letting F be an approximation of the above flux terms,

$$F^{\pm}_{j+1/2}(t) \approx f(U(x^{\pm}_{j+1/2}, t)), \tag{5.35}$$

we obtain the semi-discrete form of the finite volume scheme (4.14) as

$$\frac{d}{dt} U_j(t) + \frac{1}{\Delta x} \Big(F^-_{j+1/2}(t) - F^+_{j-1/2}(t) \Big) = 0. \tag{5.36}$$

The scheme (5.36) leads to a system of ODEs that must be integrated in time by a suitable time integration routine. Dropping the t in (5.36) for notational convenience, we can use the following numerical flux function:

$$F^-_{j+1/2} = F^+_{j+1/2} = F(U_j, U_{j+1}), \tag{5.37}$$

where F is an consistent two-point flux function like the Godunov (4.15), Rusanov (4.32) or Engquist–Osher (4.33) flux. If a standard forward Euler method is used to integrate the ODE system (5.36) in time, we obtain the standard first-order monotone finite volume scheme (4.14).

5.7. Second-order semi-discrete schemes.

The advantage of the semi-discrete formulation (5.36) lies in the fact that we can separately increase the order of spatial and temporal accuracy. The standard first-order scheme uses piecewise constants in space and a forward Euler routine for time integration. Based on our previous discussion, we know that one approach to increase the order of accuracy is to employ high-order reconstructions instead of the piecewise constant cell averages

$$U(x,t) = U_j(t) \quad \text{for } x_{j-1/2} \leqslant x < x_{j+1/2}. \tag{5.38}$$

The process of reconstructing suitable nonoscillatory (TVD) piecewise linear functions from cell averages was described in detail in the previous section. Given U_j, we can obtain the following linear function in the cell $\mathcal{C}_j$ (again dropping t for notational convenience):

$$p_j(x) = U_j + \sigma_j (x - x_j). \tag{5.39}$$

These linear functions are combined to form the piecewise linear function

$$p(x,t) = p_j(x) \quad \text{for } x_{j-1/2} \leqslant x < x_{j+1/2}. \tag{5.40}$$

In order to ensure that the reconstruction is TVD, i.e,

$$\|p\|_{BV} \leqslant \|U^{\Delta x}\|_{BV} \quad \text{for all } t,$$

we need to choose the slope σ in (5.39) suitably. From our previous discussion, we have at least two choices that satisfy the above condition: the minmod limiter (5.23) and the superbee limiter (5.26).

Last, we denote the reconstructed values at the cell interfaces as

$$U_j^+ = p_j(x_{j+1/2}), \quad U_j^- = p_j(x_{j-1/2}). \tag{5.41}$$

See Figure 5.17 for an illustration.

5.7.1. The numerical flux. The numerical flux in (5.36) is an approximation of the interface flux. If the data are represented as piecewise constant cell averages, then the two-point flux (5.37) suffices to define the numerical flux. Since we are representing the approximations as piecewise linear functions, we need to replace the cell averages in (5.37) with the relevant edge values (see Figure 5.17),

$$F_{j+1/2} = F\big(U_j^+, U_{j+1}^-\big), \tag{5.42}$$

where F is any consistent numerical flux. Since we require that the evolution be TVD, we will use monotone fluxes like the Godunov (4.15), Rusanov (4.32) and Engquist–Osher (4.33) flux. This completes the description of the second-order semi-discrete scheme (5.36).

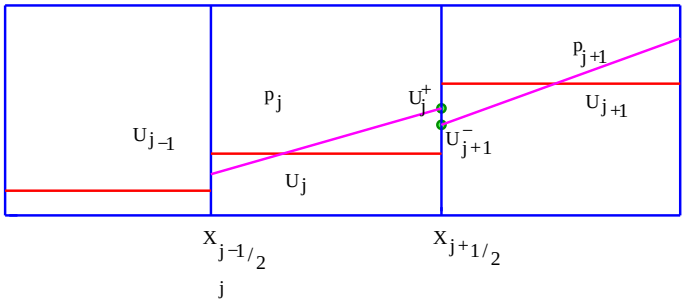


FIGURE 5.17. Second-order nonoscillatory reconstruction.

5.8. Time stepping

Denoting $U(t)$ as the vector

$$U(t) = \big[\ldots,\ U_{j-1}(t),\ U_j(t),\ U_{j+1}(t),\ \ldots\big]$$

the finite volume scheme (5.36) can be rewritten as

$$\frac{d}{dt}U(t) = \mathcal{L}(U(t)). \tag{5.43}$$

Here, the operator $\mathcal{L}$ acts pointwise on the vector U as

$$\mathcal{L}(U(t))_j := -\frac{1}{\Delta x}\big(F_{j+1/2}(t) - F_{j-1/2}(t)\big).$$

Thus, (5.43) is a system of ODEs that must be integrated in time. The simplest time integration routine is the forward Euler time integration,

$$U^{n+1} = U^n + \Delta t\mathcal{L}(U^n), \tag{5.44}$$

where $U^n = U(t^n)$ is the vector of cell averages U at time t^n. The time step Δt should be determined by a suitable CFL condition like (4.10). By choosing a TVD reconstruction in (5.39) and a monotone flux in (5.42), we can ensure that the solution update in (5.44) is TVD, i.e,

$$TV(U^{n+1}) \leqslant TV(U^n) \quad \text{for all } n. \tag{5.45}$$

However, the forward Euler method is only first-order accurate. Despite the second-order accuracy of the piecewise linear reconstruction (5.39), the first-order temporal accuracy leads to an overall first-order accuracy and negates the very purpose of using a piecewise linear reconstruction.

5.8.1. Standard Runge–Kutta method. The alternative is to employ high-order time stepping methods. These methods have been developed to a considerable extent; see standard textbooks like [**HNW87**]. The standard high-order ODE methods are of the Runge–Kutta type. The well-established standard second-order Runge–Kutta method is of the form

$$U^{n+1} = U^n + \Delta t \mathcal{L}\left(U^n + \frac{\Delta t}{2}\mathcal{L}(U^n)\right). \tag{5.46}$$

This two-stage method is second-order accurate. However it fails to satisfy the TVD requirement (5.45) and may lead to oscillatory solutions [**GST01**]. Wishing to avoid such spurious oscillations, we need to search for Runge–Kutta methods that preserve the TVD property. Such methods are termed *strong stability preserving* (SSP) Runge–Kutta methods. One second-order SSP Runge–Kutta method is

$$\begin{aligned} U^* &= U^n + \Delta t \mathcal{L}(U^n) \\ U^{**} &= U^* + \Delta t \mathcal{L}(U^*) \\ U^{n+1} &= \frac{1}{2}(U^n + U^{**}) \end{aligned} \tag{5.47}$$

We have the following stability lemma for the second-order SSP Runge–Kutta method.

Lemma 5.10. *If the discrete differential operator $\mathcal{L}$ is such that the forward Euler method* (5.44) *is TVD, then the Runge–Kutta method* (5.47) *is TVD.*

Proof. We wish to prove the TVD property (5.45) for the Runge–Kutta method (5.47). Note that this method is a combination of two forward Euler stages. Therefore,

$$TV(U^*) \leqslant TV(U^n) \quad \text{and} \quad TV(U^{**}) \leqslant TV(U^*) \leqslant TV(U^n).$$

Thus,

$$TV(U^{n+1}) = TV\left(\frac{1}{2}\big(U^n + U^{**}\big)\right) \leqslant \frac{1}{2}\big(TV(U^n) + TV(U^{**})\big) \leqslant TV(U^n).$$

□

5.9. High-resolution algorithm

With all the ingredients in place, we can state the algorithm for computing with a second-order scheme. Given cell averages U_j^n at time level t^n, we need to perform the following steps:

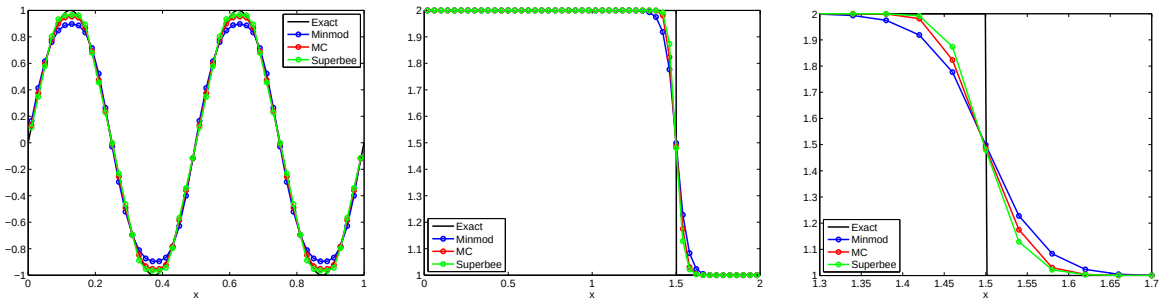
- Step 1 (Reconstruction): Given U_j, reconstruct the averages to obtain the piecewise linear function (5.39). Any nonoscillatory slope limiter like the minmod (5.23), superbee (5.26) or the MC (5.27) limiter can be used. Note that we only require the edge values $U_j^{\pm}$ (5.41) in each cell.
- Step 2 (Flux evaluation): Given the edge values $U_j^{\pm}$ in each cell, we plug these values into the numerical flux (5.42). In particular, monotone two-point fluxes like the Godunov (4.15), Engquist–Osher (4.33) and Rusanov (4.32) fluxes should be used.
- Step 3 (Time stepping): For second-order schemes, we use the second-order SSP Runge–Kutta method (5.47). As this method consists of two stages, steps 1 and 2 must be applied to each stage (e.g., U^n and U^*).

The time step Δt in (5.47) is determined by a CFL condition of the form (4.10). The second-order high resolution schemes are TVD as all the three ingredients are constructed to ensure this property.

The stencil of a second-order scheme consists of five points. This should be contrasted with the three point first-order schemes.

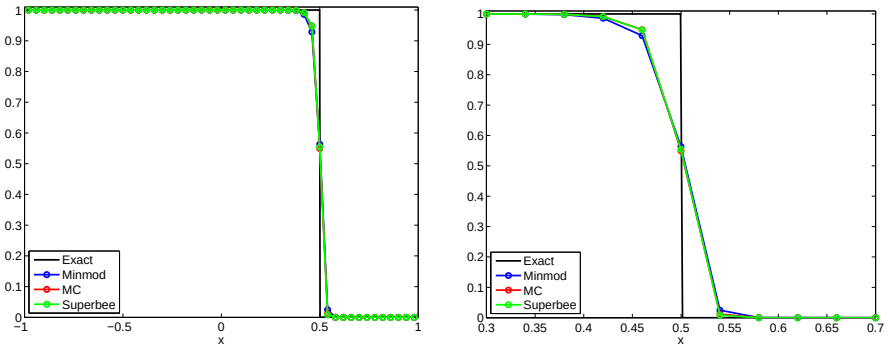
5.10. Numerical experiments

In this section, we present numerical experiments with both the linear transport equation (2.2) and Burgers' equation (3.3). To begin with, we consider the advection equation (2.2) with initial data (5.2) and periodic boundary conditions. The second-order high-resolution scheme with all three choices of slope limiters are computed and the results are displayed in Figure 5.18(a). The results are very similar to those obtained with the second-order Lax–Wendroff scheme (5.9) (see Figure 5.3). This is to be expected, as the schemes basically only differ in their time integration routines. Analogous results hold for the discontinuous initial data (2.27) and are shown in Figure 5.18(b).



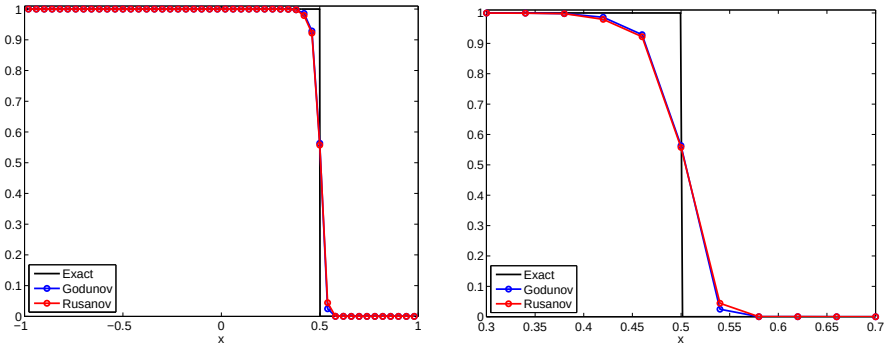
(a) Smooth data (5.2). (b) Discontinuous data (2.27). (c) Closeup of (b).

FIGURE 5.18. Linear advection equation (2.2) on a mesh of 50 cells, using various limiters.



(a) Solution at $t = 1$. (b) Closeup.

FIGURE 5.19. The Godunov scheme with different slope limiters on the problem (4.17). `[burgers_reconstr.m]`



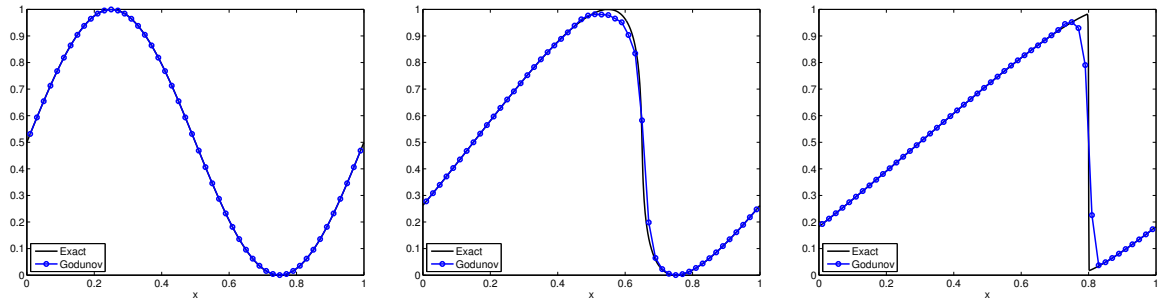
(a) Solution at $t = 1$. (b) Closeup.

FIGURE 5.20. Comparison between the Godunov and Rusanov schemes with the minmod limiter on the problem (4.17). `[burgers_reconstr.m]`

Next, we consider Burgers' equation with initial data (4.17) using the Godunov scheme. The results obtained with the high-resolution schemes are shown in Figures 5.19 and 5.20 – clearly there are very minor differences between the three choices of slope. The gain due to second-order accuracy in resolving the shock is not as much as the gain seen in the linear case.

No. of cells	Godunov		Rusanov	
	Relative error	EOC	Relative error	EOC
20	2.86	–	3.51	–
40	1.48	0.951	1.66	1.08
80	0.74	0.998	0.839	0.988
160	0.37	1	0.42	0.997
320	0.185	1	0.21	1
640	0.0925	1	0.105	1
1280	0.0463	1	0.0525	1
2560	0.0231	1	0.0263	1

TABLE 5.6. Error and order of convergence for Burgers' equation computed to time $t = 1$ with discontinuous initial conditions (4.17), using the minmod limiter. `[burgers_reconstr_OOC.m]`



(a) $t = 0$ (b) $t = 0.3$ (c) $t = 0.6$

FIGURE 5.21. Smooth initial value problem developing into a discontinuous solution. `[burgers_sine_reconstr.m]`

In Figure 5.20, we display results comparing the Godunov flux and the Rusanov flux (4.32) using the minmod limiter. While the differences in results rendered by these fluxes is pronounced at first-order, the differences at second-order are considerably smaller. Error table 5.6 shows that the rate of convergence is exactly 1. This is to be expected, as the solution is not smooth, and so the truncation error is not well defined in this case.

Last, we consider the smooth initial value problem

$$U(x,0) = \frac{1+\sin(2\pi x)}{2} \qquad \text{for } x \in [0,1]$$

with periodic boundary conditions. The solution will develop a discontinuity at time $t = -\frac{1}{\min U'(x,0)} = \frac{1}{\pi} \approx 0.318$. The solution computed with the Godunov scheme using the minmod limiter on a mesh of 50 cells is displayed in Figure 5.21. To assess the convergence rate of the scheme, we compute its experimental rate of convergence at $t = 0.3$, while the solution is still smooth. As seen in Table 5.7, the rate of convergence is now close to 2 for both the Godunov and Rusanov schemes.

No. of cells	Godunov		Rusanov	
	Relative error	EOC	Relative error	EOC
20	5.01	–	5.44	–
40	2.13	1.23	2.3	1.24
80	0.772	1.47	0.812	1.5
160	0.241	1.68	0.247	1.72
320	0.0712	1.76	0.0716	1.79
640	0.0214	1.73	0.0215	1.74
1280	0.00592	1.86	0.00592	1.86
2560	0.00168	1.82	0.00168	1.82

TABLE 5.7. Error and order of convergence for Burgers' equation for a smooth initial value problem. `[burgers_sine_reconstr_OOC.m]`

CHAPTER 6

Very high-order finite volume methods for scalar conservation laws.

The numerical examples in the preceding chapter reveal that increasing the order of accuracy of the semi-discrete finite volume scheme for approximating the scalar conservation law (3.4):

$$\frac{d}{dt}U_j(t) + \frac{1}{\Delta x}(F_{j+1/2}(t) - F_{j-1/2}(t)) = 0, \tag{6.1}$$

from one to two in both space and time led to a considerable increase in the computational efficiency, both in the smooth parts of the solution as well as near the shocks. High-resolution schemes like the limiter based schemes presented in the last chapter suffice for a large number of applications. However, there may be some situations (particularly in three space dimensions), where it is not possible to resolve very finely and even the high-resolution second-order schemes are not adequate to approximate interesting flow features. In such problems, we need to design *very high-order* schemes i.e, schemes whose formal order of accuracy is greater than two in both space and time.

In this chapter, we will present very high-order schemes. To begin with, we work with the semi-discrete form of the finite volume scheme (6.1) and focus on increasing the spatial accuracy. Then, the temporal integration is performed with high-order accurate strong stability preserving (SSP) Runge–Kutta methods.

We consider an uniform discretization of the spatial domain $[x_l, x_r]$ with mesh size Δx. The mesh points are denoted by $x_j = x_l + j\Delta x$ and they demarcate cells of the form: $\mathcal{C}_j = [x_{j-1/2}, x_{j+1/2})$. The cell average of the unknown at time t is denoted as $U_j(t)$ and we drop the t-dependence of all quantities in the sequel for notational convenience. We recall from chapters 4 and 5 that using piecewise constant cell averages results in a first-order spatially accurate scheme whereas employing piecewise linear functions (5.40) results in a second-order accurate scheme. Thus, it is natural to employ even higher order piecewise polynomial interpolations of the cell-averages in-order to obtain higher order of accuracy in space.

6.1. High-order accurate piecewise polynomial reconstructions

Consider a smooth function $V(x)$ and assume that we are given cell averages:

$$V_j = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} V(x)\,dx, \quad \forall j. \tag{6.2}$$

and would like to obtain piecewise polynomial approximations to V of degree $k \geqslant 2$. In-order to do so, we need to consider neighboring cell values in the form of a so-called *stencil* of neighboring cells. Let $r, s \geqslant 0$ be integers such that $r + s + 1 = k$, then standard approximation theory suggests that a stencil for constructing approximate polynomials of $(k-1)$-th degree are the cells,

$$\mathcal{S}_{r,j} = \{\mathcal{C}_{j-r}, \cdots, \mathcal{C}_j, \cdots, \mathcal{C}_{j+s}\}. \tag{6.3}$$

Therefore, there are k possible stencils for reconstructing a polynomial of degree $k-1$. As an example, the three possible stencils for reconstructing a piecewise quadratic functions (see Figure 6.1) are

$$\begin{aligned}
\mathcal{S}_{0,j} &= \{\mathcal{C}_j, \mathcal{C}_{j+1}, \mathcal{C}_{j+2}\},\\
\mathcal{S}_{1,j} &= \{\mathcal{C}_{j-1}, \mathcal{C}_j, \mathcal{C}_{j+1}\},\\
\mathcal{S}_{2,j} &= \{\mathcal{C}_{j-2}, \mathcal{C}_{j-1}, \mathcal{C}_j\}.
\end{aligned}$$

Note that we mark a stencil with the left shift value r as specifying r leads to an unique value for the right shift: $s = k-1-r$. For the rest of this section, we will fix r and hence the stencil $\mathcal{S}_{r,j}$ and construct an approximating polynomial $p_j^r(x)$ with the following properties,

Conservation: The approximating polynomial p_j^r is *conservative*, i.e, for all $j - r \leqslant i \leqslant j + s$, we have

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} p_j^r(x)\,dx = V_i. \tag{6.4}$$

The approximation should preserve the cell averages and is hence termed conservative.

Accuracy: Given p_j^r, we need the corner point values,

$$V_{j+1/2}^r = p_j^r(x_{j+1/2}), \quad V_{j-1/2}^r = p_j^r(x_{j-1/2}),$$

for obtaining the numerical fluxes in (6.1). These corner point values can be realized as linear combinations of the neighboring cell averages,

$$V_{j+1/2}^r = \sum_{i=0}^{k-1} c_{ri} V_{j-r+i}, \quad V_{j-1/2}^r = \sum_{i=0}^{k-1} \widehat{c}_{ri} V_{j-r+i}, \tag{6.5}$$

with c_{ri} and $\widehat{c}_{ri}$ being suitable constants. A simple argument with the ordering shows that

$$\widehat{c}_{ri} = c_{r-1,i}.$$

Therefore, the problem of finding an approximating polynomial to V reduces to determining the constants c_{ri} in (6.5). We need the approximately polynomial to be k-th order accurate:

$$V_{j+1/2}^r - V(x_{j+1/2}) = \mathcal{O}(\Delta x^k). \tag{6.6}$$

Hence, p_j^r should be of $(k-1)$-th degree.

Given point values of a smooth function, the task of constructing interpolation polynomials results from standard approximation theory. However, we are given the cell averages of the function and we need a conservative reconstruction (6.4). Standard approximation theory does not suffice and we utilize the structure of cell averages by consider the primitive of the function V,

$$\widehat{V}(x) = \int_{x_L}^{x} V(\xi)d\xi. \tag{6.7}$$

Here, x_L is any arbitrary point and we can fix it as the left boundary point x_l of the domain. Note that the cell averages of V (4.3) define point values of $\widehat{V}$ at each $x_{j+1/2}$ by

$$\widehat{V}_{j+1/2} = \widehat{V}(x_{j+1/2}) = \sum_{i=0}^{j}\int_{x_{i-1/2}}^{x_{i+1/2}} V(\xi)d\xi = \Delta x \sum_{i=0}^{j} V_j, \tag{6.8}$$

by definition (4.3). Since, the point values of $\widehat{V}$ are readily available, we will interpolate it with a polynomial of degree k from the values at points

$$x_{j-r-1/2}, \cdots, x_{j+1/2}, \cdots, x_{j+s+1/2}.$$

and call this polynomial as $P_j^r(x)$. Note that this polynomial approximates the primitive function $\widehat{V}$ to order $k+1$. Therefore setting

$$p_j^r(x) = \frac{dP_j^r}{dx}(x), \tag{6.9}$$

will approximate V to degree k as $\widehat{V}' = V$. Hence, we have shown that p_j^r satisfies the order property (6.6).

For checking the conservation property (6.4), we fix $j - r \leqslant i \leqslant j + s$ and perform the following calculations,

$$\begin{aligned}
\int_{x_{i-1/2}}^{x_{i+1/2}} p_j^r(x)\,dx &= \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{d}{dx}P_j^r(x)\,dx \quad \text{(Definition (6.9))},\\
&= P_j^r(x_{i+1/2}) - P_j^r(x_{i-1/2}) \quad \text{(Integrating by parts)},\\
&= \widehat{V}_{i+1/2} - \widehat{V}_{i-1/2} \quad \text{(Interpolation points)},\\
&= \Delta x\left(\sum_{l=0}^{i} V_l - \sum_{i=0}^{i-1} V_l\right), \quad \text{(Definition (6.8))},
\end{aligned}$$

$$= \Delta x V_i,$$

thus verifying (6.4). Hence, the interpolation polynomial p_j^r satisfies both the conservation (6.4) and order (6.6) requirements.

We do not need to explicitly calculate the interpolation polynomial (6.9) as the only information necessary in reconstructing the corner point values are the coefficients c_{ri} in (6.5) (as $\widehat{c}_{r,i} = c_{r-1,i}$). The recipe for finding the coefficients is a standard procedure from approximation theory and we refer to [**Shu97**] for a detailed account. For implementing these schemes, we provide the coefficients c_{ri} up to order $k = 4$ in the Table 6.1 below,

k	r	$i=0$	$i=1$	$i=2$	$i=3$
2	-1	$\frac{3}{2}$	$-\frac{1}{2}$		
	0	$\frac{1}{2}$	$\frac{1}{2}$		
	1	$-\frac{1}{2}$	$\frac{3}{2}$		
3	-1	$\frac{11}{6}$	$-\frac{7}{6}$	$\frac{1}{3}$	
	0	$\frac{1}{3}$	$\frac{5}{6}$	$-\frac{1}{6}$	
	1	$-\frac{1}{6}$	$\frac{5}{6}$	$\frac{1}{3}$	
	2	$\frac{1}{3}$	$-\frac{7}{6}$	$\frac{11}{6}$	
4	-1	$\frac{25}{12}$	$-\frac{23}{12}$	$\frac{13}{12}$	$-\frac{1}{4}$
	0	$\frac{1}{4}$	$\frac{13}{12}$	$-\frac{5}{12}$	$\frac{1}{12}$
	1	$-\frac{1}{12}$	$\frac{7}{12}$	$\frac{7}{12}$	$-\frac{1}{12}$
	2	$\frac{1}{12}$	$-\frac{5}{12}$	$\frac{13}{12}$	$\frac{1}{4}$
	3	$-\frac{1}{4}$	$\frac{13}{12}$	$-\frac{23}{12}$	$\frac{25}{12}$

TABLE 6.1. The coefficients c_{ri} in (6.5) up to $k = 4$

Summarizing the contents of this section, we have a described a general procedure for building up *conservative* polynomial interpolations of degree k, given the cell averages V_j of a smooth function j. Since the corner point values in each cell are the only pieces of information necessary in defining the numerical fluxes in the finite volume scheme (6.1), we have described a recipe to construct them from (6.5) and Table 6.1 up to fourth order. Higher order reconstructions can be checked from [**Shu97**] and standard books in approximation theory.

6.2. ENO reconstruction procedure

The preceding section was very general and provided a procedure for obtaining high-order interpolations once a stencil (6.3) is specified. If the function V is smooth, then any admissible stencil would yield a robust high-order approximation. The optimal stencil is then chosen to be the one with lowest approximation error. For example, the optimal 4-th order stencil leads to the following approximation,

$$V^1_{j+1/2} = -\frac{1}{12}V_{j-1} + \frac{7}{12}V_j + \frac{7}{12}V_{j+1} - \frac{1}{12}V_{j+2}.$$

However, it is well known by now that the solutions of the conservation law (3.4) are discontinuous, on account of the presence of shock waves. Using any arbitrary admissible stencil to reconstruct a discontinuous may lead to a oscillatory approximation. As an example, consider $k = 2$ and the cell averages given in example (5.21). In this case, using the upwind stencil ($r = 0$) at every cell leads to an oscillatory approximation, resulting in the increase of total variation. Similarly, using a downwind stencil ($r = 1$) at every cell also leads to an oscillatory approximation.

Entropy solution of the scalar conservation law (3.4) are TVD and it is natural to require that the approximating polynomial p^r be nonoscillatory. This is achieved in the second-order case by using limiters (Chapter 5). However, it is very hard to enforce the TVD criteria for even higher order approximations. This problem is solved by the ingenious and celebrated *ENO* procedure.

Motivation. To motivate the design of the Essentially Nonoscillatory (ENO) reconstruction procedure, we recall that the entropy solution of the conservation law (3.4) contains shocks and is discontinuous. There is no point in approximating a discontinuous function with a high-order interpolation polynomial as the approximation properties are only valid for a smooth function.

Consider the cell $\mathcal{C}_j$. The set of all admissible stencils (6.3) is specified by the left shift value $0 \leqslant r \leqslant k-1$, where k is the order of approximation. Assume that a shock exists in the neighborhood of cell $\mathcal{C}_j$. It is natural to search for those stencils in the set of admissible stencils such that they do not contain the shock. If such a *smooth* stencil exists, then the solution is smooth within this stencil and the correct order of approximation holds. Thus, we should select a stencil among the set of admissible stencils, based on smoothness of the solution. The optimal stencil should be chosen in the *direction of smoothness*.

How do we decide which of the admissible stencils is the smoothest?. A possible answer might lie in considering divided differences.

Divided differences. The reconstruction procedure involves reconstructing the primitive function $\widehat{V}$ (6.8). Define the first-order divided differences as

$$\widehat{V}[x_{j-1/2}] = \widehat{V}_{j-1/2},$$

The l-th order divided difference is defined inductively as,

$$\widehat{V}[x_{j-1/2}, \cdots, x_{i+l-1/2}] = \frac{\widehat{V}[x_{j+1/2}, \cdots, x_{j+l-1/2}] - \widehat{V}[x_{j-1/2}, \cdots, x_{j+l-3/2}]}{x_{i+l-1/2} - x_{j-1/2}}.$$

As $\widehat{V}$ is the primitive of V, we can use (6.5) to obtain

$$\widehat{V}[x_{j-1/2}, x_{j+1/2}] = \frac{\widehat{V}_{j+1/2} - \widehat{V}_{j-1/2}}{\Delta x} = V_j.$$

Hence, we can calculate the divided differences of $\widehat{V}$ in terms of the cell averages of V.

The reason for introducing divided differences lies in the fact that they define the Newton form of interpolating polynomial of degree k given by

$$P_j^r(x) = \sum_{i=0}^{k} \widehat{V}[x_{j-r-1/2}, \cdots x_{j-r+i-1/2}] \prod_{l=0}^{i-1} (x - x_{j-r+l-1/2}). \tag{6.10}$$

The divided differences are also a measure of the smoothness of the underlying function. Standard approximation theory yields that for a smooth function W,

$$W[x_{j-1/2}, \cdots, x_{j+i-1/2}] = \frac{d^i W(x)}{dx^i}(\xi), \tag{6.11}$$

for some ξ lying in the stencil and if W is discontinuous at any point in the stencil, then we have

$$W[x_{j-1/2}, \cdots, x_{j+i-1/2}] = \mathcal{O}(\frac{1}{\Delta x^i}). \tag{6.12}$$

Hence, divided differences provide some measure of the degree of smoothness of a solution. The ENO procedure utilizes divided differences to ascertain the smoothest possible stencil.

ENO algorithm. To illustrate the ENO algorithm, we describe an example involving approximation of the primitive function $\widehat{V}$ to third-order. Consequently the function V is approximated to second-order. To begin with, we require that the cell $\mathcal{C}_j$ should be involved in the reconstruction. This implies that we start with a two-point stencil,

$$\mathcal{S}_j^2 = \{x_{j-1/2}, x_{j+1/2}\}, \tag{6.13}$$

to reconstruct $\widehat{V}$. The Newton form (6.10) yields the piecewise linear function,

$$P_j^1(x) = \widehat{V}[x_{j-1/2}] + \widehat{V}[x_{j-1/2}, x_{j+1/2}](x - x_{j-1/2}). \tag{6.14}$$

In order to obtain a quadratic interpolation, we need to add another point to the stencil. There are two possibilities: either we consider the left neighboring point $x_{j-3/2}$ leading to the stencil

$$\mathcal{S}_j^3 = \{x_{j-3/2}, x_{j-1/2}, x_{j+1/2}\}, \tag{6.15}$$

and quadratic interpolation,

$$\begin{aligned} P^2(x) &= \widehat{V}[x_{j-1/2}] + \widehat{V}[x_{j-1/2}, x_{j+1/2}](x - x_{j-1/2}) + \widehat{V}[x_{j-3/2}, x_{j-1/2}, x_{j+1/2}](x - x_{j-1/2})(x - x_{j+1/2}), \\ &= P^1(x) + \widehat{V}[x_{j-3/2}, x_{j-1/2}, x_{j+1/2}](x - x_{j-1/2})(x - x_{j+1/2}), \end{aligned} \tag{6.16}$$

or we can add the right neighboring point $x_{j+3/2}$ leading to the stencil

$$\widetilde{\mathcal{S}}_j^3 = \{x_{j-1/2}, x_{j+1/2}, x_{j+3/2}\}, \tag{6.17}$$

and quadratic interpolation,

$$\begin{aligned} \widetilde{P}^2(x) &= \widehat{V}[x_{j-1/2}] + \widehat{V}[x_{j-1/2}, x_{j+1/2}](x - x_{j-1/2}) + \widehat{V}[x_{j-1/2}, x_{j+1/2}, x_{j+3/2}](x - x_{j-1/2})(x - x_{j+1/2}), \\ &= P^1(x) + \widehat{V}[x_{j-1/2}, x_{j+1/2}, x_{j+3/2}](x - x_{j-1/2})(x - x_{j+1/2}). \end{aligned} \tag{6.18}$$

Note that the only difference between the reconstructed polynomials lies in the quadratic term $(x - x_{i-1/2})(x - x_{i+1/2})$, which is multiplied by two different divided differences for each stencil. So choosing stencils $\mathcal{S}_j^3$ or $\widetilde{\mathcal{S}}_j^3$ boils down to choosing between the two second-order divided differences. From the preceding discussion, we recall that divided differences are an indicator of smoothness of function $\widehat{V}$ and choose the divided difference with the least magnitude i.e, if

$$|\widehat{V}[x_{j-3/2}, x_{j-1/2}, x_{j+1/2}]| \leqslant |\widehat{V}[x_{j-1/2}, x_{j+1/2}, x_{j+3/2}]|, \tag{6.19}$$

then $\mathcal{S}_j^3$ is chosen as the stencil to reconstruct the quadratic interpolation. Otherwis $\widetilde{\mathcal{S}}_j^3$ is chosen.

Note that

$$\begin{aligned} \widehat{V}[x_{j-3/2}, x_{j-1/2}, x_{j+1/2}] &= \frac{V_j - V_{j-1}}{\Delta x}, \\ \widehat{V}[x_{j-1/2}, x_{j+1/2}, x_{j+3/2}] &= \frac{V_{j+1} - V_j}{\Delta x}. \end{aligned}$$

Hence (6.19) amounts to choosing a *limiter* for the adjacent slopes. This should be compared with the limiters presented in Chapter 5.

If we want to construct a cubic interpolation, we have to add another point to the current stencil, say $\mathcal{S}_j^3$. We have two choices: either the left neighbor $x_{j-5/2}$ or the right neighbor $x_{j+3/2}$. Again the modulus of the third-order divided differences,

$$|\widehat{V}[x_{j-5/2}, x_{j-3/2}, x_{j-1/2}, x_{j+1/2}]|, \quad |\widehat{V}[x_{j-3/2}x_{j-1/2}, x_{j+1/2}, x_{j+3/2}]|$$

are compared and the one with minimum modulus is chosen to be added to the stencil. This procedure can be iterated k times to obtain a preferred stencil. The ENO algorithm is

Step 1. Given the cell averages of a function V and the order of the desired interpolation polynomial k, compute divided differences of the primitive function $\widehat{V}$ using cell averages V_j.

Step 2. For the cell $\mathcal{C}_j$, define the two-point stencil:

$$\mathcal{S}_j^2 = \{x_{j-1/2}, x_{j+1/2}\}.$$

For all $m = 2, 3, \cdots, k$, assume that m-th stencil is known and takes the form,

$$\mathcal{S}_j^m = \{x_{i-1/2}, x_{i+1/2}, \cdots, x_{i+m-1/2}\},$$

for some i, Then if

$$|\widehat{V}[x_{i-3/2}, x_{i-1/2}, \cdots, x_{i+m-1/2}]| \leqslant |\widehat{V}[x_{i-1/2}, x_{i+1/2}, \cdots, x_{i+m+1/2}]|$$

we add $x_{i-3/2}$ to the stencil to form the new stencil

$$\mathcal{S}_j^{m+1} = \{x_{i-3/2}, x_{i-1/2}, x_{i+1/2}, \cdots, x_{i+m-1/2}\},$$

else, we add $x_{i+m+1/2}$ to the stencil to form the new stencil .

Step 3. The above procedure determines the stencil uniquely. Denote the stencil as $\mathcal{S}_{r,j}$ by the left shift value r and use (6.5) to determine the corner point values $V^r_{j+1/2}$ and $V^r_{j-1/2}$. The coefficients c_{rj} can be checked from Table 6.1.

The above algorithm is easy to code and results in a *nonoscillatory* reconstruction of V from its cell averages. Note that the primitive function $\widehat{V}$ need not be computed at any stage as only its divided differences (computed from cell averages) are required. A graphical illustration of the ENO procedure is depicted in Figure 6.1.

We illustrate the ENO procedure by considering the Heaviside function V, with cell averages given by (5.21). The aim is to obtain a third-order piecewise quadratic interpolation at cell $\mathcal{C}_{J-1}$. Using the ENO algorithm leads to the following set of stencils,

$$\begin{aligned}
\mathcal{S}_j^2 &= \{x_{J-3/2}, x_{J-1/2}\},\\
\mathcal{S}_j^3 &= \{x_{J-5/2}, x_{J-3/2}, x_{J-1/2}\},\\
\mathcal{S}_j^4 &= \{x_{J-7/2}, x_{J-5/2}, x_{J-3/2}, x_{J-1/2}\},
\end{aligned}$$

Thus the reconstruction is based completely from the left. Similarly, the ENO reconstruction at $\mathcal{C}_J$ uses a stencil, takes values completely from the right. The reconstructed function is thus the the original function and the entire reconstruction is TVD.

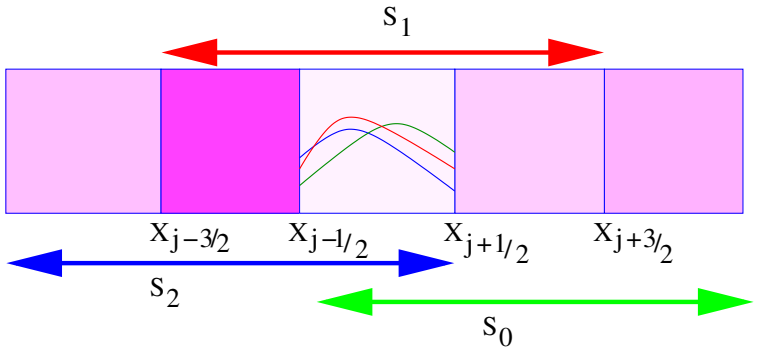


FIGURE 6.1. A graphical description of the ENO procedure: The aim is reconstruct a second-order polynomial in the cell $\mathcal{C}_j$. The three candidate stencils are denoted by S_0 (green), S_1 (red) and S_2 (blue). The corresponding quadratic functions are depicted inside the cell. The ENO procedure selects the stencil with the smoothest approximation. In this case, the blue stencil S_2 is selected and the corresponding polynomial (in blue) is used to obtain the approximate corner point values

Properties of the ENO reconstruction. The ENO reconstruction, as outlined above has the following properties,

i. Let $P_j(x)$ be an k-degree interpolation polynomial of the primitive function $\widehat{V}(x)$, based on the ENO procedure. Then, it satisfies the accuracy condition,

$$P_j(x) - \widehat{V}(x) = \mathcal{O}(\Delta x^{k+1}), \tag{6.20}$$

for all j provided that the cell $\mathcal{C}_j$ does not contain a discontinuity i.e, the function V is sufficiently smooth. The proof of this fact is a straightforward consequence of the construction of the interpolation polynomial.

ii. If $\mathcal{C}_j$ contains a discontinuity of $\widehat{V}$, then $P_j(x)$ is monotone in the cell . The proof is non-trivial and interested readers can consult [**HOEC86**] for a rigorous proof.

iii. The reconstruction is Total Variation Bounded (TVB), in fact there exists a function W such that

$$W(x) - P_j(x) = \mathcal{O}(\Delta x^{k+1}),$$

such that

$$\|W\|_{TV} \leqslant \|\widehat{V}\|_{TV}.$$

The proof is a consequence of properties (i) and (ii). Define $W = \widehat{V}$ if the cells are smooth and $W = P_j$ if the cell has a discontinuity. Monotonicity automatically implies the TVB property.

Problems with the ENO approximation. The ENO approximation is very robust and leads to efficient nonoscillatory approximations of conservation laws (this will be discussed in detail in the sequel). However, it has two major drawbacks,

1. The ENO procedure can lead to a non-smooth and abrupt change of stencils at neighboring mesh points. Even round-off errors in divided differences can lead to changes of stencils, causing a lack of smoothness for the numerical flux in (6.1).
2. To obtain a k-th order ENO approximation, we need to consider all the candidate stencils consisting of $(2k-1)$ points and choose the "smoothest" stencil among them. Standard approximation theory establishes that a $(2k-1)$-th order approximation can be constructed from $(2k-1)$ interpolation points. Thus, we are not utilizing the optimal reconstruction property of the underlying stencil.

These limitations, particularly the one concerning sub-optimal accuracy encourage the modification of the ENO framework to design the WENO or *Weighted essential nonoscillatory* framework.

6.3. WENO Reconstruction

Consider a function V, expressed in terms of its cell averages V_j. Let $k \geqslant 2$ be the order of approximation and $0 \leqslant r \leqslant k-1$. Define $s = k-1-r$ and denote the stencil $\mathcal{S}_{r,j}$ by (6.3). For any fixed r and stencil $\mathcal{S}_{r,j}$, Section 6.1 provides a general recipe for using the primitive function $\widehat{V}$ to obtain a k-th order accurate and conservative approximation of V. In particular, the point values $V^r_{j+1/2}$ and $V^r_{j-1/2}$ are defined in terms of (6.5). Thus every stencil leads to unique k-th order accurate approximation of the point values.

In the ENO procedure, we choose the *smoothest* stencil among all the k candidate stencils and define the corresponding point values. The WENO procedure differs from this stencil selection. It utilizes a standard result from approximation theory which states that if V is smooth in all the candidate stencils, then there exists constants c_r such that the function,

$$\widetilde{V}_{j+1/2} = \sum_{r=0}^{k-1} d_r V^r_{j+1/2} = V(x_{j+1/2}) + \mathcal{O}(\Delta x^{2k-1}). \tag{6.21}$$

Thus, the k-candidate stencils based on $2k-1$ interpolation points lead to a $(2k+1)$-th order accurate interpolation. For $k=2$, the constants are

$$d_0 = \frac{2}{3}, \quad d_1 = \frac{1}{3}. \tag{6.22}$$

Similarly for $k = 3$, the constants are

$$d_0 = \frac{3}{10}, \quad d_1 = \frac{3}{5}, \quad d_2 = \frac{1}{10}. \tag{6.23}$$

The constants for even higher order approximations can be readily obtained. It is easy to check that $d_r \geqslant 0$ and for consistency,

$$\sum_{r=0}^{k-1} d_r = 1.$$

Similarly,

$$\widetilde{V}_{j-1/2} = \sum_{r=0}^{k-1} \widetilde{d}_r V_{j-1/2}^r = V(x_{j-1/2}) + \mathcal{O}(\Delta x^{2k-1}). \tag{6.24}$$

By symmetry, we obtain that $\widetilde{d}_r = d_{k-1-r}$.

The point values are thus weighted averages of the reconstruction from each candidate stencil. Using the point values $\widetilde{V}_{j+1/2}$ will lead to an oscillatory reconstruction as the function V can contain discontinuities. Hence, we need to combine the idea of weighted averages of reconstructions from candidate stencil together with a procedure for identifying stencils that are smooth and ensuring that they contribute more to the average than stencils that contain discontinuities. This balance or choice of weights lies at the heart of the WENO approach.

WENO reconstruction is based on the weighted point value,

$$V_{j+1/2} = \sum_{r=0}^{k-1} \omega_r V_{j+1/2}^r, \tag{6.25}$$

We need that $\omega_r \geqslant 0$ and

$$\sum_{r=0}^{k-1} \omega_r = 1,$$

hold for the sake of consistency.

If V were a smooth function, then weights ω have to be chosen in-order ensure that the approximation (6.25) is $(2k+1)$-th order accurate. This is indeed the case if the weights satisfy,

$$\omega_r = d_r + \mathcal{O}(\Delta x^{k-1}). \tag{6.26}$$

Then,

$$\begin{aligned}
\sum_{r=0}^{k-1} \omega_r V_{j+1/2}^r - \sum_{r=0}^{k-1} d_r V_{j+1/2}^r, &= \sum_{r=0}^{k-1} \omega_r V_{j+1/2}^r - \sum_{r=0}^{k-1} d_r V_{j+1/2}^r + V(x_{j+1/2}) - V(x_{j+1/2}) \\
&= \sum_{r=0}^{k-1} \omega_r V_{j+1/2}^r - \sum_{r=0}^{k-1} d_r V_{j+1/2}^r + V(x_{j+1/2}) \left(\sum_{r=0}^{k-1} \omega_r - \sum_{r=0}^{k-1} d_r \right), \text{ (consistency)}, \\
&= \sum_{r=0}^{k-1} (\omega_r - d_r)(V_{j+1/2}^r - V(x_{j+1/2})), \\
&= \sum_{r=0}^{k-1} \mathcal{O}(\Delta x^{k-1}) \mathcal{O}(\Delta x^k), \quad \text{by (6.26) and (6.6)}, \\
&= \mathcal{O}(\Delta x^{2k-1}).
\end{aligned}$$

Combining the above inequality with (6.21) and (6.25), we obtain that

$$V_{j+1/2} - V(x_{j+1/2}) = \mathcal{O}(\Delta x^{2k-1}), \tag{6.27}$$

thus satisfying the desired order of accuracy.

As discussed before, the weights ω_r should reflect the smoothness of the corresponding stencil, marked by r. A clever choice of computationally efficient weights that measure smoothness leads to

$$\omega_r = \frac{\alpha_r}{\sum_{m=0}^{k-1} \alpha_m}, \tag{6.28}$$

where

$$\alpha_r = \frac{d_r}{(\beta_r + \varepsilon)^2}, \tag{6.29}$$

with ε being a very small tolerance to ensure that the denominator is always non-zero. The key to finding robust weights is to design β_r carefully so that both the accuracy requirement (6.26) is met and the weight reflects the smoothness of the stencil.

If the function V was smooth in the stencil marked by r, then the smoothness indicator β_r should be

$$\beta_r = \mathcal{O}(\Delta x^2).$$

Similarly, if the stencil under consideration contained a discontinuity of V, then the smoothness indicator β_r is chosen to be

$$\beta_r = \mathcal{O}(1)$$

Using the above choices, we observe that the weight ω_r in (6.28) becomes,

$$\omega_r = \mathcal{O}(1),$$

if the function V is smooth in the stencil. Similarly, if the stencil contains a discontinuity then the weight is

$$\omega_r = \mathcal{O}(\Delta x^4).$$

This choice of β ensures that a stencil containing the smooth parts of a function has a much greater weight in the reconstruction (6.25) than a stencil that contains a discontinuity. Thus, the WENO procedure replicates some aspects of the ENO procedure and weighs stencils, based on their smoothness.

The task of finding smoothness indicators β is highly non-trivial. A clever choice in [**LOC94**] is based on the following recipe: consider a stencil $\mathcal{S}_{r,j}$ and denote the $k-1$-th degree polynomial interpolating V as $p_r(x)$, then β_r is given by

$$\beta_r = \sum_{m=1}^{k-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x^{2m-1} \left(\frac{d^m p_r(x)}{dx^m} \right)^2 dx. \tag{6.30}$$

The scaling of Δx ensures that the resulting derivatives do not depend on Δx. The above expression is a square of the L^2 norms of all derivatives of the interpolation polynomial, up to $k-1$. This integral can be explicitly computed in the special case of $k = 2$ and we obtain,

$$\beta_0 = (V_{j+1} - V_j)^2, \quad \beta_1 = (V_j - V_{j-1})^2. \tag{6.31}$$

Similarly, we can compute (6.30) when $k = 3$ to obtain,

$$\begin{aligned}
\beta_0 &= \frac{13}{12}(V_j - 2V_{j+1} + V_{j+2})^2 + \frac{1}{4}(3V_j - 4V_{j+1} + V_{j+2})^2, \\
\beta_1 &= \frac{13}{12}(V_{j-1} - 2V_j + V_{j+1})^2 + \frac{1}{4}(V_{j-1} - V_{j+1})^2, \\
\beta_2 &= \frac{13}{12}(V_{j-2} - 2V_{j-1} + V_j)^2 + \frac{1}{4}(V_{j-2} - 4V_{j-1} + 3V_j)^2,
\end{aligned} \tag{6.32}$$

Higher order smoothness indicators are harder to compute explicitly. We summarize the complete algorithm below:

6.4. WENO Algorithm

Step 1. Given the cell averages of a function V and a desired approximation of order k, denote $0 \leqslant r \leqslant k-1$ as the left shift and $\mathcal{S}_{r,j}$ (6.3) as the corresponding stencil. Construct point values $V^r_{j+1/2}, V^r_{j-1/2}$ by (6.5) for all r.

Step 2. Compute the coefficients d_r and $\widetilde{d}_r$ from (6.21) and (6.24). For $k = 2, 3$, we can use the explicit values (6.22) and (6.23).

Step 3. Compute smoothness indicators β_r using (6.30). Explicit values for $k = 2$ and $k = 3$ are given by (6.31) and (6.32).

Step 4: Define ω_r from (6.28), (6.29). Similarly define,

$$\widetilde{\omega}_r = \frac{\widetilde{\alpha}_r}{\sum_{m=0}^{k-1} \widetilde{\alpha}_m}, \quad \widetilde{\alpha}_r = \frac{\widetilde{d}_r}{(\beta_r + \varepsilon)^2},$$

Step 5: Compute the $(2k+1)$-th accurate point values,

$$V_j^+ = \sum_{r=0}^{k-1} \omega_r V_{j+1/2}^r, \quad V_j^- = \sum_{r=0}^{k-1} \widetilde{\omega}_r V_{j-1/2}^r. \tag{6.33}$$

A graphical description of the WENO procedure is provided in Figure 6.2.

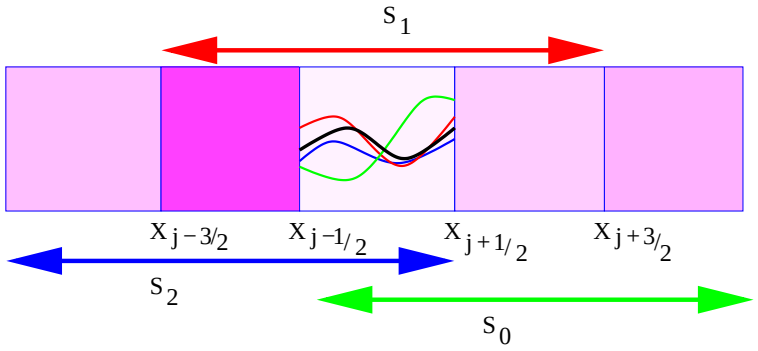


FIGURE 6.2. A graphical description of the WENO procedure: The aim is reconstruct a second-order polynomial in the cell $\mathcal{C}_j$. The three candidate stencils are denoted by S_0 (green), S_1 (red) and S_2 (blue). The corresponding quadratic functions are depicted inside the cell. The WENO procedure computes an weighted average of the three polynomials resulting in the polynomial shown in black, which is used to obtain the approximate corner point values

Remark 6.1. *The implementation of a WENO interpolation is easier to code than an ENO interpolation and requires fewer if statements. Consequently, it is more efficient in terms of run time.*

Remark 6.2. *We would like to point out that a k-order WENO interpolation results in a $(2k-1)$-th order accuracy of the approximation. Therefore, a second-order WENO procedure results in third-order of accuracy and a third-order WENO procedure leads to fifth-order of accuracy.*

To illustrate the WENO procedure, we again consider the numerical example with cell averages given by (5.21). Let $k = 2$. There are two candidate stencils,

$$\mathcal{S}_{0,J} = \{\mathcal{C}_{J-1}, \mathcal{C}_J\} \quad \text{and} \quad \mathcal{S}_{1,J} = \{\mathcal{C}_J, \mathcal{C}_{J+1}\}.$$

The corresponding point values are given by (6.5) as

$$V_{j+1/2}^0 = -\frac{1}{2}, \quad V_{j+1/2}^1 = 0.$$

The coefficients d are given by (6.22) and the smoothness indicators (6.30) are

$$\beta_0 = 1, \quad \beta_1 = 0.$$

We can compute the weights ω by (6.28), (6.29) as

$$\omega_0 = \frac{2\varepsilon^2}{1 + 2\varepsilon^2} \approx 0, \qquad \omega_1 = 1 - \omega_0 \approx 1.$$

Therefore, the reconstructed value (6.33) is

$$V_J^+ \approx 0.$$

Similarly, we can compute,

$$V_J^- \approx 0, \quad V_{J-1}^+ \approx 1, \quad V_{J-1}^- \approx 1,$$

and observe that the resulting reconstruction is TVD.

6.5. Numerical flux calculation

The ENO and WENO approximations as described above are very general and can be applied to obtain nonoscillatory reconstructions for any function. However, we are interested in computing approximate solutions of the scalar conservation law (3.4). Therefore, we work with cell averages $U_j(t)$ at any given time t.

Given the cell averages at any time level, we use either the ENO or the WENO procedures to obtain the corner point values,

$$U_j^-(t) \approx U(x_{j-1/2}+,t), \quad U_j^+(t) \approx U(x_{j+1/2}-,t).$$

Both the ENO and WENO procedures result in a *nonoscillatory* choice of the above point values. Once these point values are specified, the numerical flux in (6.1) can be determined as

$$F_{j+1/2}(t) = F(U_j^+(t), U_{j+1}^-(t)). \tag{6.34}$$

Here F can be any consistent, monotone two-point flux function like the Godunov (4.15), Engquist-Osher (4.33) or Rusanov flux (4.32). This completes the description of a semi-discrete finite volume scheme (6.1).

6.6. Time-Stepping

The finite volume scheme (6.1) is semi-discrete and needs to be updated in time with a suitable time stepping scheme. In Chapter 5, we discussed the issue of time-stepping extensively and introduced the concept of *Strong Stability Preserving* (SSP) Runge–Kutta methods. These methods are designed to ensure that the updates are TVD if the underlying Forward Euler step is TVD. A second-order two stage SSP Runge–Kutta method was described in (5.47). Since our spatial order of approximation, constructed with suitable ENO and WENO schemes is more than second-order accurate, we need to construct even high order SSP Runge–Kutta methods.

Let $\mathcal{L}$ be update operator defined in (5.43) for a numerical flux (6.34) defined by a ENO or WENO procedure, then we can rewrite the semi-discrete finite volume scheme (6.1) as a system of ODEs (5.43) in terms of $\mathcal{L}$. A general k-stage Runge–Kutta method is of the form,

$$\begin{aligned} U^{(m)} &= \sum_{l=0}^{m-1} \left(\alpha_{ml} U^{(l)} + \Delta t \beta_{ml} \mathcal{L}\left(U^{(l)}\right)\right), \quad 1 \leqslant m \leqslant k \\ U^{(0)} &= U^n, \quad U^{(k)} = U^{n+1}, \end{aligned} \tag{6.35}$$

with the coefficients α_{ml} and β_{ml}. If the coefficients $\alpha_{ml}, \beta_{ml} \geqslant 0$, it is clear that $U^{(m)}$ is just a convex combination of Forward Euler steps. Hence, any good properties of the forward-Euler method are inherited by the corresponding Runge–Kutta method. In particular, for the second-order Runge–Kutta method (5.47), we have that

$$\begin{aligned} \alpha_{10} = 1, \quad \alpha_{20} = \frac{1}{2}, \quad \alpha_{21} = \frac{1}{2}, \\ \beta_{10} = 1, \quad \beta_{20} = 0, \quad \beta_{21} = \frac{1}{2}, \end{aligned}$$

ensuring that this method is TVD.

A third-order three stage TVD method for (5.43) takes the form,

$$\begin{aligned} U^{(1)} &= U^n + \Delta t \mathcal{L}(U^n), \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\mathcal{L}(U^{(1)}), \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\mathcal{L}(U^{(2)}). \end{aligned} \tag{6.36}$$

It is not possible to obtain a fourth-order SSP Runge–Kutta method that has positive β_{ml}'s. One has use a suitable adjoint operator to ensure that a SSP method can be designed. Interested reader can

consult [**Shu97**] for description of such a method. For practical purposes, it suffices to use the third-order Runge–Kutta method (6.36). In case an even higher order time integration method is required, we can use the standard four stage Runge Kutta method given by,

$$
\begin{aligned}
U^{(1)} &= \Delta t \mathcal{L}(U^n), \\
U^{(2)} &= \Delta t \mathcal{L}(U^n + \frac{1}{2}U^{(1)}), \\
U^{(3)} &= \Delta t \mathcal{L}(U^n + \frac{1}{2}U^{(2)}), \\
U^{(4)} &= \Delta t \mathcal{L}(U^n + U^{(3)}), \\
U^{n+1} &= U^n + \frac{1}{6}U^{(1)} + \frac{1}{3}U^{(2)} + \frac{1}{3}U^{(3)} + \frac{1}{6}U^{(4)}.
\end{aligned} \tag{6.37}
$$

Although (6.37) may not be TVD, it is observed to be quite robust in practice. This completes the description of a fully discrete high-order finite volume scheme.

6.7. Numerical Experiments

We will test the very high-order schemes of this chapter for the linear advection equation (2.2) as well as Burgers' equation (3.3). We denote the following schemes:

ENO2	Second-order ENO scheme with second-order Runge–Kutta time stepping (5.47).
ENO3	Third-order ENO scheme with third-order Runge–Kutta time stepping (6.36).
ENO4	Fourth-order ENO scheme with fourth-order Runge–Kutta time stepping (6.37).
WENO3	Third-order WENO scheme with second-order Runge–Kutta time stepping (5.47).
WENO5	Fifth-order WENO scheme with third-order Runge–Kutta time stepping (6.36).

To begin with, we consider the advection equation (2.2) with initial data (5.2) and periodic boundary conditions. The results with all the five schemes , with Godunov (upwind) flux for 100 mesh points for time $t = 5$ are presented in Figure **??**. For the sake of comparison, we present results comparing the first-order upwind scheme also. The errors are present in Table **??** show the correct rates of convergence.

We repeat the same experiments for Burgers' equation with initial data (4.17) and (5.2).

CHAPTER 7

Linear hyperbolic systems in one space dimension

In the last few chapters, we have treated the nonlinear scalar conservation law (3.4) and have designed robust high-order finite volume schemes to approximate its entropy solutions.

Despite their occurance in many interesting models, scalar conservation laws are too simplistic to be used for modeling complex physical phenomena as these problems involve the interaction of many unknowns. These problems (see Chapter 1 for examples) are modeled by a nonlinear system of conservation laws:

$$\mathbf{U}_t + \mathbf{f}(\mathbf{U})_x = 0, \tag{7.1}$$

where $\mathbf{U} = [U^1, U^2, \ldots, U^m]^\top$ is the vector of unknowns and $\mathbf{f} = [f^1, f^2, \ldots, f^m]^\top$ is the flux vector. Note that (7.1) represents an $m \times m$ system of conservation laws in one space dimension. Multi-dimensional systems will be considered later. The system (7.1) is supplemented with suitable initial and boundary conditions. When $m = 1$, the system reduces to the scalar conservation law (3.4).

To begin with, we consider the simplest case of (7.1) in the form a linear system,

$$\mathbf{U}_t + A\mathbf{U}_x = 0. \tag{7.2}$$

Here, A is an $m \times m$ matrix with constant (in both space and time) entries. Linear systems arise for instance when *linearizing* (7.1) around a constant state $\overline{\mathbf{U}}$, which amounts to solving (7.2) with the constant matrix $A = \frac{\partial \mathbf{f}}{\partial \mathbf{U}}(\overline{\mathbf{U}})$.

A related linear system with variable coefficients takes the form

$$\mathbf{U}_t + (A(x,t)\mathbf{U})_x = 0. \tag{7.3}$$

It can be recast into the non-conservative form

$$\mathbf{U}_t + A(x,t)\mathbf{U}_x = -A_x(x,t)\mathbf{U}. \tag{7.4}$$

7.1. Examples of linear systems

7.1.1. The wave equation. The simplest example of (7.2) is given by the one-dimensional wave equation,

$$u_{tt} - c^2 u_{xx} = 0. \tag{7.5}$$

This equation is also known as the string equation, as it models vibrations in media like strings and rods. Its derivation can be found in many text books; see for instance [**TW09**]. The wave equation can be written as a first-order system by defining auxiliary variables $v = cu_x$ and $w = -u_t$. With this *change of variables*, it is easy to show that (7.5) transforms to

$$\begin{aligned} v_t + cw_x &= 0 \\ w_t + cv_x &= 0. \end{aligned} \tag{7.6}$$

Hence, the wave equation (7.6) is an example of the linear system (7.2) with

$$\mathbf{U} = \begin{bmatrix} v \\ w \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & c \\ c & 0 \end{bmatrix}. \tag{7.7}$$

7.1.2. Maxwell's equations. The dynamics of electromagnetic waves is modeled by the Maxwell equations of electrodynamics:

$$\begin{aligned} \mathbf{B}_t + \operatorname{curl}(\mathbf{E}) &= 0 \\ \mathbf{E}_t - c^2 \operatorname{curl}(\mathbf{B}) &= -\frac{\mathbf{j}}{\varepsilon_0} \\ \operatorname{div}(\mathbf{B}) &= 0 \\ \operatorname{div}(\mathbf{E}) &= \frac{\rho}{\varepsilon_0}. \end{aligned} \tag{7.8}$$

Here, $\mathbf{B} = [B^1, B^2, B^3]^\top$ and $\mathbf{E} = [E^1, E^2, E^3]^\top$ are the magnetic and electric fields, respectively. The charge and current density are denoted by ρ and $\mathbf{j}$, respectively. The speed of light is given by $c^2 = \mu_0 \varepsilon_0$, where μ_0 and ε_0 are constants. In one space dimension, solving for electromagnetic waves propagating in the x-direction (with no current) reduces (7.8) to

$$\begin{aligned} B^2_t + E^3_x &= 0 \\ E^3_t + c^2 B^2_x &= 0. \end{aligned} \tag{7.9}$$

Thus, Maxwell's equations are equivalent to the wave equation (7.7) with $\mathbf{U} = [cB^2, E^3]^\top$, leading to another concrete application of the wave equation.

7.1.3. Linearized Euler equations. The Euler equations of gas dynamics (1.10) are a nonlinear system of conservation laws in several spatial dimensions. Let ρ, u and p denote the density, velocity and pressure of the gas, respectively. Restricting the equations to one space dimension and linearizing (1.10) around a constant state $\overline{\mathbf{U}} = [\bar\rho, \bar u, \bar p]^\top$, we obtain the *linearized Euler equations*,

$$\begin{aligned} \rho_t + \bar u \rho_x + \bar\rho u_x &= 0 \\ u_t + \bar u u_x + \frac{1}{\bar\rho} p_x &= 0 \\ p_t + \gamma \bar p u_x + \bar u p_x &= 0. \end{aligned} \tag{7.10}$$

This system of equations can be written as a linear system (7.2) with

$$\mathbf{U} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix}, \qquad A = \begin{bmatrix} \bar u & \bar\rho & 0 \\ 0 & \bar u & \frac{1}{\bar\rho} \\ 0 & \gamma \bar p & \bar u \end{bmatrix}. \tag{7.11}$$

The linearized Euler equations are a good approximation of the more complex Euler equations (1.10) when the solution only consists of small perturbations around constant states.

Given the above examples, the study of the linear system (7.2) is important in its own right. Furthermore, it serves as an example of the nonlinear system (7.1), and so the study of the linear system will be a starting point in the design of efficient numerical schemes for approximating (7.1).

7.2. Hyperbolicity and characteristic decomposition

We will consider the initial value problem associated with (7.2) by augmenting it with initial data

$$\mathbf{U}(x, 0) = \mathbf{U}_0(x). \tag{7.12}$$

One of the key notions underlying the behavior of the solutions of (7.2) is the concept of *hyperbolicity*. For scalar equations, the notion of hyperbolicity referred to the property of finite speed of propagation. Such a notion will hold true for a system.

Definition 7.1 (Hyperbolic system)**.** *The linear system* (7.2) *is* hyperbolic *if the matrix A is diagonalizable and has m real eigenvalues. Similarly, the equation* (7.3) *is hyperbolic if the matrix $A(x,t)$ is diagonalizable and has m real eigenvalues for all x, t. Both systems are termed* strictly hyperbolic *if the eigenvalues are distinct.*

This definition can be extended to the nonlinear case (7.1) *if the flux $\mathbf{f}$ is at least C^1 in $\mathbf{U}$. We will say that* (7.1) *is hyperbolic if for any state $\mathbf{U}$ in the range of $\mathbf{f}$, the Jacobian matrix $\nabla_{\mathbf{U}} \mathbf{f}$ is diagonalizable in $\mathbb{R}$. We return to nonlinear systems in the next chapter.*

Thus, the concept of hyperbolicity for a system like (7.2) is tied to the eigenstructure of the underlying matrix. Why is the eigenstructure linked to finite speed of propagation? The answers lies in diagonalizability. A matrix A is diagonalizable if it has a complete set of eigenvectors, i.e, there exists m linearly independent vectors $r_1, \dots, r_m \in \mathbb{R}^m$ and corresponding eigenvalues $\lambda_1, \dots, \lambda_m \in \mathbb{R}$ such that

$$Ar_p = \lambda_p r_p.$$

We can then define the matrix of eigenvectors

$$R = \begin{bmatrix} r_1 \,|\, r_2 \,|\, \cdots \,|\, r_m \end{bmatrix} \tag{7.13}$$

and the diagonal matrix of eigenvalues

$$\Lambda = \operatorname{diag}\big(\lambda_1, \dots, \lambda_m\big) \tag{7.14}$$

and use them to *diagonalize* the underlying matrix A – that is, write it in the form

$$A = R\Lambda R^{-1}. \tag{7.15}$$

The diagonalization of A leads us to the following decoupling:

$$\begin{aligned}
\mathbf{U}_t + A\mathbf{U}_x &= 0 \\
\mathbf{U}_t + R\Lambda R^{-1}\mathbf{U}_x &= 0 && \text{(from (7.15))} \\
\mathbf{U}_t + R\Lambda \left(R^{-1}\mathbf{U}\right)_x &= 0 && \text{(constant coefficient)} \\
\left(R^{-1}\mathbf{U}\right)_t + R^{-1}R\Lambda \left(R^{-1}\mathbf{U}\right)_x &= 0 && \text{(multiplying both sides with } R^{-1}\text{)} \\
\left(R^{-1}\mathbf{U}\right)_t + \Lambda \left(R^{-1}\mathbf{U}\right)_x &= 0.
\end{aligned}$$

Introducing the vector of characteristic variables

$$\mathbf{W} = R^{-1}\mathbf{U},$$

the above calculation leads to

$$\mathbf{W}_t + \Lambda \mathbf{W}_x = 0. \tag{7.16}$$

Thus, the hyperbolic linear system (7.2) is completely decoupled into a set of scalar linear transport equations, obtained by writing (7.16) componentwise:

$$W^p_t + \lambda_p W^p_x = 0 \qquad \text{for } p = 1, \dots, m, \tag{7.17}$$

where $W^p(x,t)$ is the p-th component of $\mathbf{W}(x,t)$. This decoupled set of transport equations can be solved explicitly by the method of characteristics (see Chapter 2) to obtain

$$W^p(x,t) = W^p_0\left(x - \lambda_p t\right), \tag{7.18}$$

where

$$\mathbf{W}_0(x) = R^{-1}\mathbf{U}_0(x)$$

and again $W^p_0(x,t)$ is the p-th component of $\mathbf{W}_0(x,t)$. Therefore, we can explicitly write the initial value problem for a hyperbolic system (7.2) as

$$\mathbf{U}(x,t) = R\mathbf{W}(x,t). \tag{7.19}$$

Remark 7.2. *One may wonder where Definition 7.1 comes from. In fact, considering a linear system, this is the only way to achieve that the initial value problem associated with* (7.2) *subjected to the initial conditions* (7.12) *is well posed in* L^2*. Let us first recall the notion of well-posedness in the sense of Hadamard.*

Since we are looking for L^2 *solution, we can use the Fourier transform and the problem becomes:*

$$\frac{\partial \hat{\mathbf{U}}}{\partial t}(k,t) + ik \cdot A\hat{\mathbf{U}}(k,t) = 0,$$

that can easily be integrated:

$$\hat{\mathbf{U}}(k,t) = e^{ik\cdot At}\hat{\mathbf{U}}_0.$$

So the question is to look at the L^2 *of the operator*

$$e^{ik\cdot At}.$$

Since A is real valued, its eigenvalues consist of pairs of complex conjugates. The real part of the eigenvalues of $ik \cdot A$ are related to the imaginary part of the eigenvalues of A. For each eigenvalue, there exists an eigenvector, and for sure, for any pair of conjugate eigenvalues, one has a negative imaginary part. If at least one of the eigenvalues of A has a non zero imaginary part, then the norm of the operator cannot be bounded uniformly in time. This shows that a necessary condition for stability is to have real eigenvalues.

Now let is show that the matrix is diagonalizable. If not, there is one Jordan block. We can assume without problem that A is this Jordan block. Taking the vector $x = (0, 0, \dots, 1)^T$ and computing $e^{ik\cdot At}x$, we get

$$e^{i\lambda kt}\left(\frac{(itk)^{n-1}}{(n-1)!}, \frac{(itk)^{n-2}}{(n-2)!}, \dots, 1\right)$$

which L^2 norm is

$$\left(\sum_{p=0}^{n-1}\left(\frac{(tk)^p}{k!}\right)^2\right)^{1/2}$$

which cannot be bounded. Hence, for the L^2 norm of $e^{ik\cdot At}$ to be bounded, we need A to be diagonalizable in $\mathbb{R}$.

The converse is immediate.

7.3. Solutions of Riemann problems, waves

We illustrate the explicit solution (7.19) for the Riemann initial data

$$\mathbf{U}_0(x) = \begin{cases} \mathbf{U}_L & \text{if } x < 0 \\ \mathbf{U}_R & \text{if } x > 0. \end{cases} \tag{7.20}$$

The first step in finding an explicit solution is to determine Λ and R for the given system A. These are used to define $\mathbf{W}_0 = R^{-1}\mathbf{U}_0$, which will be of the form

$$\mathbf{W}_0(x) = \begin{cases} \mathbf{W}_L & \text{if } x < 0 \\ \mathbf{W}_R & \text{if } x > 0. \end{cases}$$

with $\mathbf{W}_L = R^{-1}\mathbf{U}_L$ and $\mathbf{W}_R = R^{-1}\mathbf{U}_R$. Then we can solve (7.17) in terms of the transport equation (7.18) to obtain

$$W^p(x,t) = \begin{cases} W_L^p & \text{if } x < \lambda_p t \\ W_R^p & \text{if } x > \lambda_p t. \end{cases}$$

The solution $\mathbf{U}$ is obtained by letting $\mathbf{U} = R\mathbf{W}$. This solution consists of m waves, one for each eigenvector. The p-th wave propagates with a speed of λ_p, and across this wave, there is a jump in only the p-th component of $\mathbf{W}$. This is seen clearly by decomposing the jump $\mathbf{U}_R - \mathbf{U}_L$ in terms of eigenvectors:

$$\begin{aligned} \mathbf{U}_R - \mathbf{U}_L &= R(\mathbf{W}_R - \mathbf{W}_L) \\ &= \sum_{p=1}^{m} (W_R^p - W_L^p) r_p \\ &= \sum_{p=1}^{m} \alpha^p r_p, \end{aligned} \tag{7.21}$$

where $\alpha^p := W_R^p - W_L^p$ is the *wave strength* of the p-th wave. Hence, the jump across the p-th wave is proportional to the eigenvector r_p. Thus, the solution of the Riemann problem for a linear system corresponds to decomposing the initial jump into m waves (with the corresponding eigenvalue being the wave speed), with the jump across the p-th wave being proportional to the corresponding eigenvector.

7.3.1. Example: The wave equation. We consider the linear wave equation (7.6). It is straightforward to find that the eigenvalues and eigenvectors of A are

$$\Lambda = \operatorname{diag}(-c, c), \qquad R = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix}, \qquad R^{-1} = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} \end{bmatrix}. \tag{7.22}$$

Thus, the wave equation is strictly hyperbolic and diagonalizable, so the above construction of the explicit solution is applicable. Consider the Riemann problem (7.20) for the wave equation (7.6). Let $c = 1$ for simplicity. The solution of the Riemann problem is then

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \text{if } x < -t \\ \mathbf{U}_* & \text{if } -t < x < t \\ \mathbf{U}_R & \text{if } x > t, \end{cases} \tag{7.23}$$

where the mid-state $\mathbf{U}_*$ is given by

$$\mathbf{U}_* = R \begin{bmatrix} \frac{v_R - u_R}{2} \\ \frac{v_L + u_L}{2} \end{bmatrix}.$$

Thus, it is straightforward to calculate solutions to the Riemann problem for the wave equation.

7.3.2. Example: Linearized Euler equations. The eigenstructure of the linearized Euler equations (7.10) can be explicit computed as

$$\Lambda = \operatorname{diag}\left(\bar{u} - \bar{a},\ \bar{u},\ \bar{u} + \bar{a}\right), \qquad R = \begin{bmatrix} -\frac{\bar{\rho}}{\bar{a}} & 1 & \frac{\bar{\rho}}{\bar{a}} \\ 1 & 0 & 1 \\ -\bar{\rho}\bar{a} & 0 & \bar{\rho}\bar{a} \end{bmatrix}, \tag{7.24}$$

where

$$\bar{a} = \sqrt{\frac{\gamma \bar{p}}{\bar{\rho}}}$$

denotes the local sound speed. Hence, the linearized Euler equations are strictly hyperbolic. The two waves corresponding to r_1 and r_3 are termed the *acoustic* or *sound waves* and the wave corresponding to r_2 is called a *contact wave*. The Riemann problem for the linearized Euler equations can be solved explicitly with the procedure outlined above.

7.4. Finite volume schemes

In this section, we construct finite volume schemes to approximate the linear system (7.2). For simplicity, we consider a uniform grid in both space and time with mesh size Δx and time step Δt. The mesh size and time step are related via a CFL type condition to be specified in the sequel. We denote the grid points as $x_{j+1/2} = x_L + (j + 1/2)\Delta x$ and the time levels by t^n. The domain is divided into cells denoted by

$$\mathcal{C}_j = [x_{j+1/2}, x_{j-1/2}).$$

See Figure 4.1 for an illustration of the grid.

By now, we are familiar with the design procedure (see Chapter 4) and know that it consists of the following steps:

(i) **Reconstruction:** Realize the cell averages $\{\mathbf{U}_j^n\}_{j\in\mathbb{Z}}$ as the piecewise constant function

$$\mathbf{U}^n(x) = \mathbf{U}_j^n \qquad \text{for } x \in \mathcal{C}_j.$$

(ii) **Evolution:** Evolve the data to the next time level by solving

$$\begin{aligned} \mathbf{U}_t + A\mathbf{U}_x &= 0 \qquad x \in \mathbb{R},\ t > t^n \\ \mathbf{U}(x, t^n) &= \mathbf{U}^n(x) \qquad x \in \mathbb{R}. \end{aligned} \tag{7.25}$$

This amounts to solving a series of Riemann problems

$$\begin{aligned} &\mathbf{U}_t + A\mathbf{U}_x = 0, \qquad x \in \mathbb{R},\ t > t^n \\ &\mathbf{U}(x_{j+1/2}, t^n) = \begin{cases} \mathbf{U}_j^n & \text{if } x < x_{j+1/2}, \\ \mathbf{U}_{j+1}^n & \text{if } x_{j+1/2} < x \end{cases} \end{aligned} \tag{7.26}$$

in the vicinity of each cell interface $x_{j+1/2}$. These Riemann problems can be solved exactly (as described in the previous section) or approximately (see the sequel). We know that the fastest wave speed for each problem is bounded by the modulus of the maximum eigenvalue,

$$\lambda_{\max} = \max_{1 \leqslant p \leqslant m} |\lambda_p|,$$

where λ_p is the p-th eigenvalue of A. Hence, imposing a CFL condition

$$\lambda_{\max} \frac{\Delta t}{\Delta x} \leqslant \frac{1}{2} \tag{7.27}$$

prevents the waves from the neighboring problems to interact before time level t^{n+1}.

(iii) **Averaging:** Average the solution of (7.25),

$$\mathbf{U}_j^{n+1} := \frac{1}{\Delta x} \int_{\mathcal{C}_j} \mathbf{U}\big(x, t^{n+1}\big), \qquad j \in \mathbb{Z}.$$

Integrating (7.26) over $\mathcal{C}_j \times [t^n, t^{n+1})$ gives the following explicit expression for $\mathbf{U}_j^{n+1}$:

$$\mathbf{U}_j^{n+1} = \mathbf{U}_j^n - \frac{\Delta t}{\Delta x} \left(\mathbf{F}_{j+1/2}^n - \mathbf{F}_{j-1/2}^n \right) \tag{7.28}$$

with

$$\mathbf{F}_{j+1/2}^n = \mathbf{F}\left(\mathbf{U}_j^n, \mathbf{U}_{j+1}^n\right) := A\mathbf{U}_{j+1/2}\big(x_{j+1/2}, t^n + 0\big). \tag{7.29}$$

Here, $\mathbf{U}_{j+1/2}$ is the solution of the (exact or approximate) solution of the Riemann problem (7.26). We remark that the solution is self-similar and its value at the interface is time independent.

The finite volume scheme (7.28) is similar to the scheme for scalar equations. The only difference lies in the more complicated structure of solutions of the Riemann problem (7.26). We need to complete (7.28) by computing the numerical flux explicitly.

7.4.1. Godunov flux. The Riemann problem (7.26) can be solved explicitly as described before. The resulting flux evaluation (7.29) defines the Godunov flux for the linear system (7.2). It turns out that we can derive a neat expression for the Godunov flux.

Let $a^+ = \max(a, 0)$ and $a^- = \min(a, 0)$. Then $a = a^+ + a^-$ and $|a| = a^+ - a^-$ for any $a \in \mathbb{R}$. The explicit solution of the Riemann problem (7.26) can be computed and the interface value $\mathbf{U}_{j+1/2}(x_{j+1/2}, 0) = \mathbf{U}^*$ can be calculated (see (7.21)) by

$$\mathbf{U}^* = \mathbf{U}_j^n + \sum_{p:\lambda_p<0} \alpha_{j+1/2}^p r_p, \tag{7.30}$$

where $\alpha_{j+1/2}^p$ is the wave strength of the p-th wave, given by the p-th component of $R^{-1}(\mathbf{U}_{j+1}^n - \mathbf{U}_j^n)$. Since we are interested in calculating the interface flux in (7.29), we multiply both sides of (7.30) by A to obtain

$$\begin{aligned} A\mathbf{U}^* &= A\mathbf{U}_j^n + A\Big(\sum_{p:\lambda_p<0} \alpha_{j+1/2}^p r_p \Big) \\ &= A\mathbf{U}_j^n + \sum_{p:\lambda_p<0} \alpha_{j+1/2}^p A r_p \\ &= A\mathbf{U}_j^n + \sum_{p:\lambda_p<0} \alpha_{j+1/2}^p \lambda_p r_p \qquad (\text{as } r_p \text{ are eigenvectors of } A) \\ &= A\mathbf{U}_j^n + \sum_p \lambda_p^- \alpha_{j+1/2}^p r_p. \end{aligned} \tag{7.31}$$

Similarly, we have

$$\mathbf{U}^* = \mathbf{U}_{j+1}^n - \sum_{p:\lambda_p \geqslant 0} \alpha_{j+1/2}^p r_p.$$

Repeating the calculations of (7.31), we obtain

$$A\mathbf{U}^* = A\mathbf{U}^n_{j+1} - \sum_p \lambda_p^+ \alpha^p_{j+1/2} r_p. \tag{7.32}$$

Taking the average of (7.31) and (7.32) leads to

$$\begin{aligned} A\mathbf{U}^* &= \frac{1}{2}\left(A\mathbf{U}^n_j + A\mathbf{U}^n_{j+1} - \sum_p (\lambda_p^+ - \lambda_p^-)\alpha^p_{j+1/2} r_p \right) \\ &= \frac{1}{2} A\left(\mathbf{U}^n_j + \mathbf{U}^n_{j+1}\right) - \frac{1}{2}\sum_p |\lambda_p| \alpha^p_{j+1/2} r_p. \end{aligned} \tag{7.33}$$

Using the definition of the wave strength $\alpha^p_{j+1/2}$ and defining

$$|\Lambda| = \mathrm{diag}(|\lambda_1|, \cdots, |\lambda_m|),$$

it is straightforward to rewrite (7.33) as

$$\mathbf{F}^n_{j+1/2} = A\mathbf{U}^* = \frac{1}{2} A\left(\mathbf{U}^n_j + \mathbf{U}^n_{j+1}\right) - \frac{1}{2} R|\Lambda| R^{-1}\left(\mathbf{U}^n_{j+1} - \mathbf{U}^n_j\right). \tag{7.34}$$

This is the explicit form of the Godunov flux for a linear system.

The Godunov scheme (7.28) with flux (7.34) for the linear system (7.2) should be compared with the upwind flux (2.17) for the linear transport equation (2.2). In particular, both fluxes are written in the central + diffusion form. The upwinding for a system is more complicated and involves characteristic decomposition. Note that the flux (7.34) reduces to the standard upwind flux for a scalar equation, when $m = 1$.

7.4.2. Lax-Friedrichs and Rusanov flux. The Godunov flux (7.34) requires a characteristic decomposition of the system. Explicit expressions for the eigenvalues and eigenvectors may not be available in some cases, and in others it might be computationally costly to evaluate them. Hence, there is scope for designing cheaper alternative numerical fluxes that are easy to implement. Approximate Riemann solvers (see Chapter 4) provide a recipe for designing such fluxes. The key idea is to replace the exact solution of the Riemann problem (7.26) with an approximate solution. The Lax-Friedrichs flux is based on a two-wave approximate Riemann solver (see Section 4.2.2). It is straightforward to generalize it to the case of a linear system: We replace the exact solution (consisting of m waves) with exactly two waves, one moving to the left and another to the right (see Figure 4.7). The wave speeds are $-\frac{\Delta x}{2\Delta t}$ and $\frac{\Delta x}{2\Delta t}$, which is the maximum possible wave speed consistent with the CFL condition (7.27). The resulting flux is

$$\mathbf{F}^n_{j+1/2} = \mathbf{F}^{\mathrm{LxF}}\left(\mathbf{U}^n_j, \mathbf{U}^n_{j+1}\right) = \frac{1}{2} A\left(\mathbf{U}^n_j + \mathbf{U}^n_{j+1}\right) - \frac{\Delta x}{2\Delta t}\left(\mathbf{U}^n_{j+1} - \mathbf{U}^n_j\right). \tag{7.35}$$

(compare to (4.30)). This flux is trivial to implement as it does not require any characteristic information.

As in the scalar case, the Lax-Friedrichs scheme ((7.28) with (7.35)) is likely to be diffusive. A more reasonable choice of wave speeds in the approximate Riemann solver is the largest eigenvalue of A. (Note that the maximum wave speed in the Riemann problem (7.26) is always bounded by the maximum eigenvalue of A.) The resulting flux is the Rusanov flux

$$\mathbf{F}^n_{j+1/2} = \mathbf{F}^{\mathrm{Rus}}\left(\mathbf{U}^n_j, \mathbf{U}^n_{j+1}\right) = \frac{1}{2} A\left(\mathbf{U}^n_j + \mathbf{U}^n_{j+1}\right) - \frac{\lambda_{\max}}{2}\left(\mathbf{U}^n_{j+1} - \mathbf{U}^n_j\right) \tag{7.36}$$

(compare to (4.32)). The only characteristic information used in the Rusanov scheme is an estimate on the maximum eigenvalue of A.

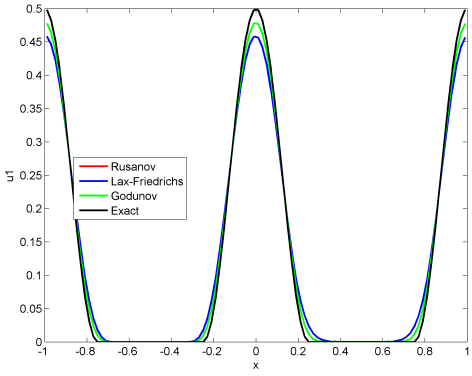
7.5. Numerical experiments

In this section we present numerical experiments for linear systems (7.2) using the Godunov, Lax-Friedrichs and Rusanov schemes.

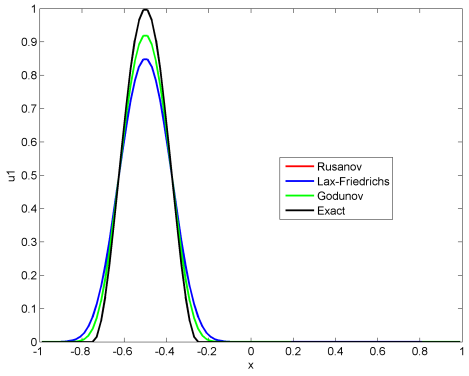
7.5.1. Numerical experiment 1. Consider the wave equation (7.6) with wave speed $c = 1$. Let the initial data be the Riemann problem

$$u_0(x) = \begin{cases} 1 - \cos^2(2\pi(x - 0.25)) & \text{if } 0.25 \leqslant x \leqslant 0.75 \\ 0 & \text{otherwise} \end{cases} \tag{7.37}$$
$$v_0(x) = 0.$$

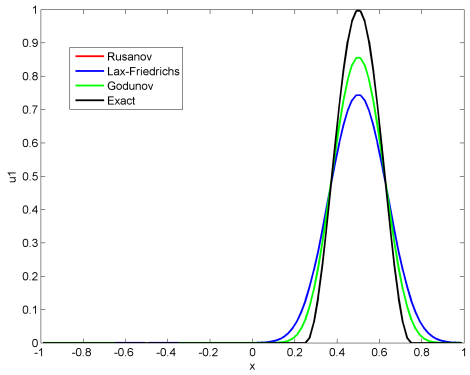
We compute on the periodic domain $x \in [-1, 1]$ and divide the domain into 100 grid cells. In Figure 7.1 we present the computational results with the Godunov, Lax-Friedrichs and Rusanov schemes at time $t = 0.5$, 1 and 2. The initial profile for u first separates into two parts, and at time $t = 1$ they meat at $x = -0.5$ and merge together. This process is again repeated at time $t = 2$, at which time the solution is the same as at $t = 0$.



(a) $t = 0.5$



(b) $t = 1$



(c) $t = 2$

FIGURE 7.1. Wave equation (7.6) computed at time $t = 0.5$, 1 and 2 with smooth initial conditions (7.37) and periodic boundary conditions on 100 cells using Rusanov, Lax-Friedrichs and Godunov fluxes. `[waveBump.m]`

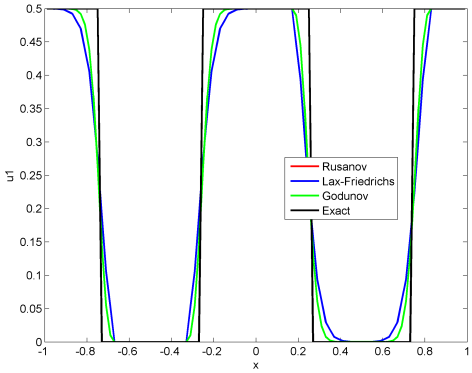
From Figure 7.1 it is clear that the Lax-Friedrichs scheme is more diffusive than the Godunov and Rusanov schemes. On the other hand, the Godunov and Rusanov schemes produce an identical profile. The reason for this is that in this case the Rusanov and Godunov schemes are identical.

Exercise 7.3. *Show that the Godunov and Rusanov fluxes* (7.34) *and* (7.36) *are identical for the wave equation* (7.7).

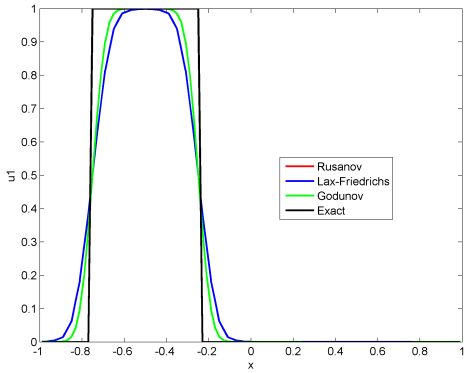
7.5.2. Numerical experiment 2. Consider the wave equation (7.6) with wave speed $c = 1$ and initial data

$$u_0(x) = \begin{cases} 1 & \text{if } 0.25 \leqslant x \leqslant 0.75 \\ 0 & \text{otherwise} \end{cases} \qquad v_0(x) = 0. \tag{7.38}$$

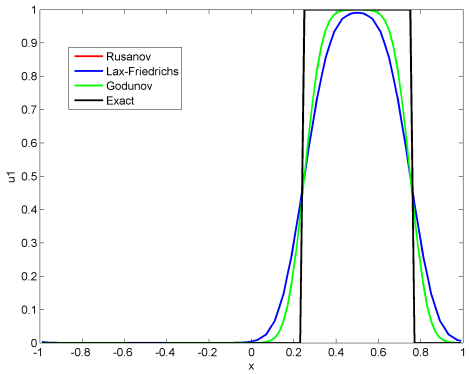
We compute on the periodic domain $x \in [-1, 1]$ and divide the domain into 100 grid cells. Results are



(a) $t = 0.5$



(b) $t = 1$



(c) $t = 2$

FIGURE 7.2. Wave equation (7.6) computed at time $t = 0.5$, 1 and 2, with discontinuous initial conditions (7.38) and periodic boundary conditions on 100 cells using Rusanov, Lax-Friedrichs and Godunov fluxes. `[waveDisc.m]`

presented in Figure 7.2. The results are again plotted at time $t = 0.5$, 1 and 2. The solution behaves similar to the smooth case. Again we see that the Lax-Friedrichs scheme is the most diffusive and that the Godunov and Rusanov schemes produce identical results.

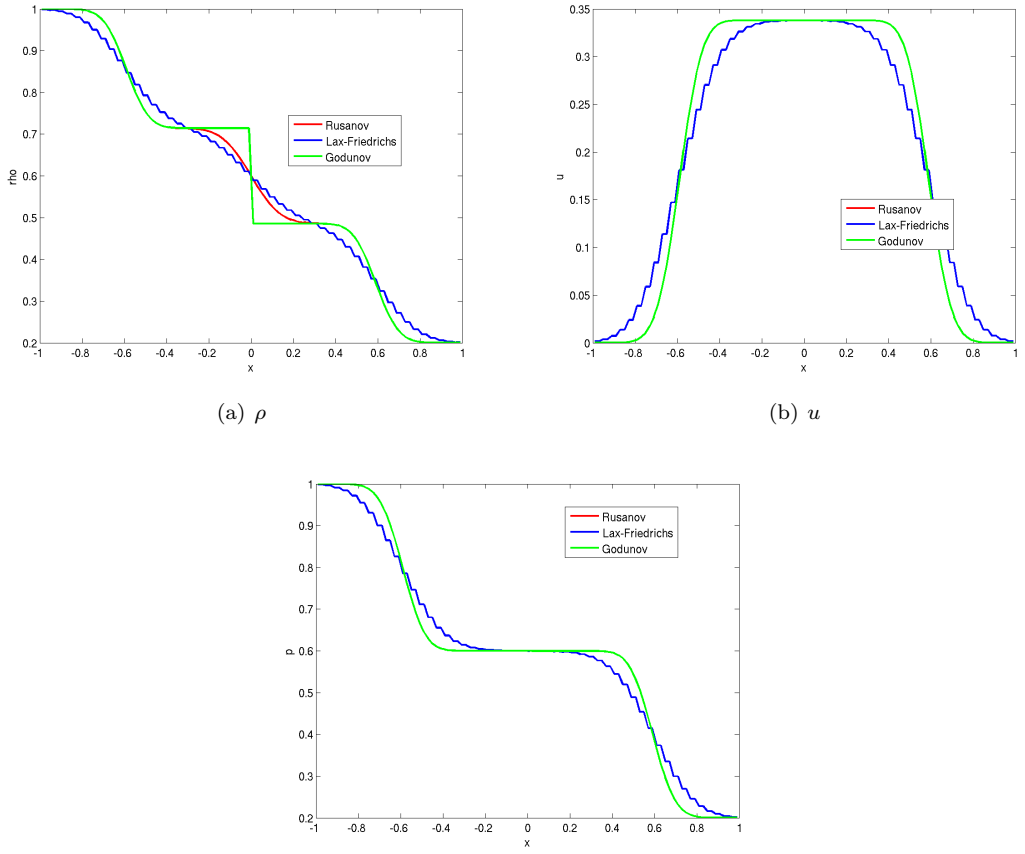
7.5.3. Numerical experiment 3. Consider the linearized Euler equation (7.10) with parameters

$$\begin{bmatrix} \bar{\rho} \\ \bar{u} \\ \bar{p} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \qquad \gamma = \frac{7}{5}$$

and initial data

$$
\begin{aligned}
\rho_0(x) &= \begin{cases} 1 & \text{if } x < 0 \\ 0.2 & \text{if } x > 0 \end{cases} \\
u_0(x) &= 0 \\
p_0(x) &= \begin{cases} 1 & \text{if } x < 0 \\ 0.2 & \text{if } x > 0 \end{cases}
\end{aligned}
\tag{7.39}
$$

in the computational domain $[-1, 1]$ with Neumann (outflow) type boundary conditions. In Figure 7.3 we plot the density, velocity and pressure at time $t = 0.5$. The solution for density contain three waves, one for each eigenvalue. The fastest wave is moving with speed equal to the highest eigenvalue of the system. Note that the velocity and pressure contain only two waves.



(a) ρ

(b) u

(c) p

FIGURE 7.3. Linearized Euler equation (7.10) computed up to time $t = 0.5$ with initial conditions (7.39) and outflow boundary conditions. `[linEuler.m]`

The most striking feature in Figure 7.3 is that the Godunov solver resolves the stationary wave in density *exactly*. This is a well-known feature of the Godunov scheme. In the velocity and pressure solutions, the Rusanov and Godunov schemes compute the exact same solutions. Of the three schemes, the Lax-Friedrichs scheme is by far the most inaccurate one.

7.6. High-order finite volume schemes

It is relatively straightforward to extend the second-order schemes of Chapter 5 to the linear system (7.2). Following Chapter 5, we use the semi-discrete form of the finite volume scheme (see (5.36)),

$$\frac{d}{dt}\mathbf{U}_j(t) + \frac{1}{\Delta x}\left(\mathbf{F}_{j+1/2}(t) - \mathbf{F}_{j-1/2}(t)\right), \tag{7.40}$$

where we denote the cell average as

$$\mathbf{U}_j(t) = \frac{1}{\Delta x}\int_{x_{j-1/2}}^{x_{j+1/2}} \mathbf{U}(x,t)\,dx.$$

Given cell averages $\mathbf{U}_j$, we wish to obtain a linear reconstruction $\mathbf{p}_j$ in each cell $\mathcal{C}_j$ of the form

$$\mathbf{p}_j(x) = \mathbf{U}_j + \sigma_j(x - x_j). \tag{7.41}$$

These linear functions are combined to form the piecewise linear function

$$\mathbf{p}(x,t) = \mathbf{p}_j(x) \qquad \text{for } x_{j-1/2} \leqslant x < x_{j+1/2}. \tag{7.42}$$

Contrary to the situation in Chapter 5, the slope σ_j is now a vector, and so the theory for scalar equations does not translate directly to the present situation. There are two basic approaches to constructing the slope σ_j.

7.6.1. Reconstruction in primitive variables. One approach to constructing σ_j is to reconstruct in primitive (conserved) variables U_j^p $(p = 1, \dots, m)$. This involves applying the reconstruction procedures of Chapter 5 to each component of $\mathbf{U}_j$, thus constructing σ_j component by component.

7.6.2. Reconstruction in characteristic variables. An alternative reconstruction involves using the characteristic variables. Given the cell averages $\mathbf{U}_j$, we define the corresponding characteristic average $\mathbf{W}_j = R^{-1}\mathbf{U}_j$. We can then perform a componentwise reconstruction in $\mathbf{W}_j$, as outlined in the previous section, to obtain a reconstruction

$$\mathbf{W}_j(x) = \mathbf{W}_j + \mu_j(x - x_j). \tag{7.43}$$

The primitive variables are then obtained from

$$\mathbf{p}_j(x) = R\mathbf{W}_j(x). \tag{7.44}$$

7.6.3. The numerical flux. As in the scalar case, the numerical flux $\mathbf{F}$ in (7.40) is defined by

$$\mathbf{F}_{j+1/2} = \mathbf{F}\left(\mathbf{U}_j^+, \mathbf{U}_{j+1}^-\right), \tag{7.45}$$

where

$$\mathbf{U}_j^+ = \mathbf{p}_j(x_{j+1/2}), \qquad \mathbf{U}_j^- = \mathbf{p}_j(x_{j-1/2}), \tag{7.46}$$

and $\mathbf{F}$ is either the Godunov (7.34), Lax-Friedrichs (7.35) or Rusanov (7.36) numerical flux.

7.6.4. Time stepping. The time integration for (7.40) is performed with an SSP Runge-Kutta method, as described in Chapter 5. Denoting

$$\mathbf{U}(t) = [\dots, \mathbf{U}_{j-1}(t), \mathbf{U}_j(t), \mathbf{U}_{j+1}(t), \dots],$$

the finite volume scheme (7.40) can be rewritten as

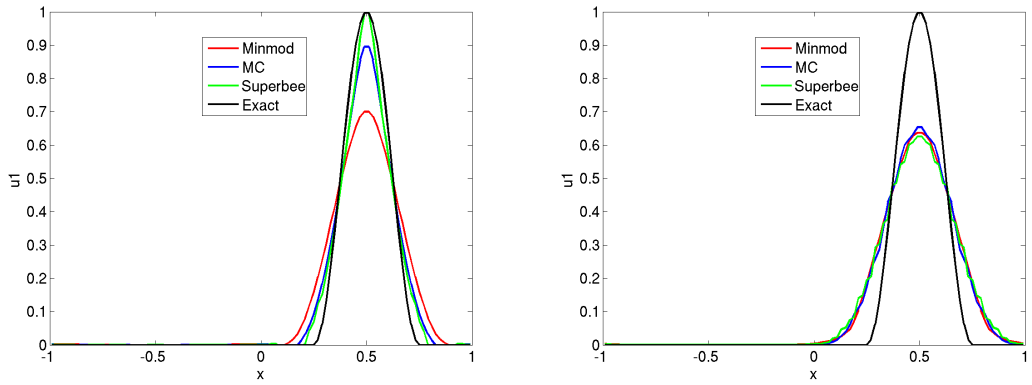
$$\frac{d}{dt}\mathbf{U}(t) = \mathcal{L}(\mathbf{U}(t)), \tag{7.47}$$

where the operator $\mathcal{L}$ acts pointwise on the vector U as

$$\mathcal{L}(\mathbf{U}(t))_j := -\frac{1}{\Delta x}\left(\mathbf{F}_{j+1/2}(t) - \mathbf{F}_{j-1/2}(t)\right).$$

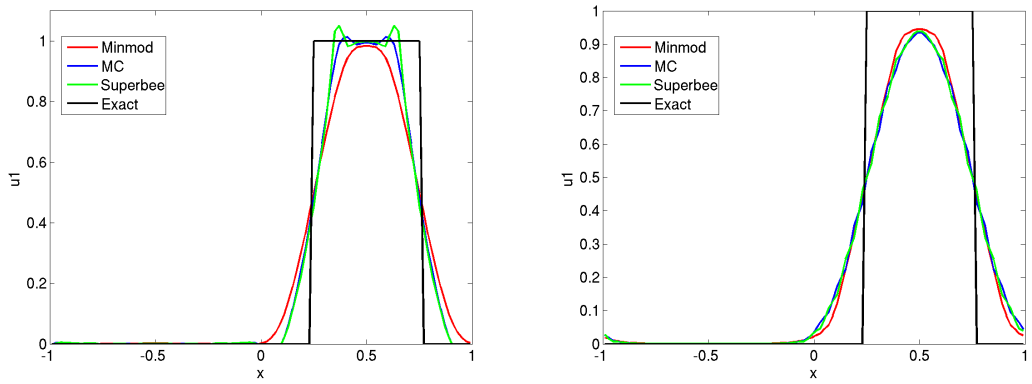
The second-order SSP Runge-Kutta method is then defined by

$$\begin{aligned} \mathbf{U}^* &= \mathbf{U}^n + \Delta t\mathcal{L}(\mathbf{U}^n) \\ \mathbf{U}^{**} &= \mathbf{U}^* + \Delta t\mathcal{L}(\mathbf{U}^*) \\ \mathbf{U}^{n+1} &= \frac{1}{2}(\mathbf{U}^n + \mathbf{U}^{**}). \end{aligned} \tag{7.48}$$



(a) Componentwise reconstruction (b) Characteristic reconstruction

FIGURE 7.4. Wave equation (7.6) computed up to time $t = 1$, with smooth initial conditions (7.37) and periodic boundary conditions on a mesh of 100 grid cells. Computed with the Godunov flux using the minmod, superbee and MC limiters. `[waveBump_2nd.m]`



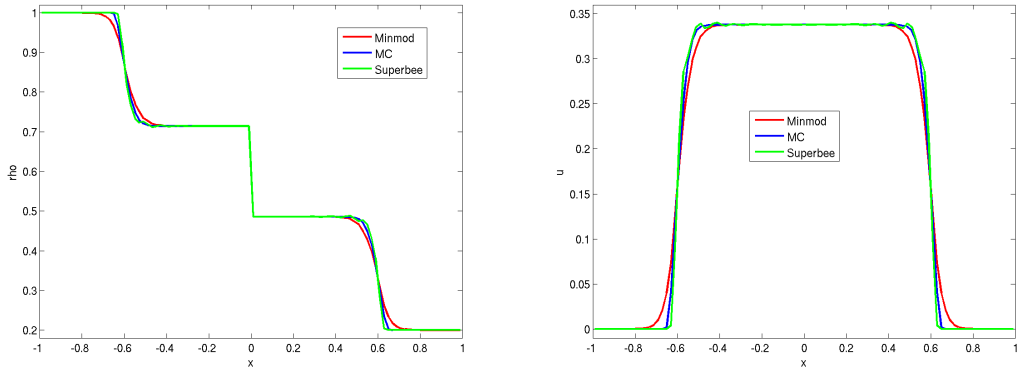
(a) Componentwise reconstruction (b) Characteristic reconstruction

FIGURE 7.5. Wave equation (7.6) computed up to time $t = 1$, with discontinuous initial conditions (7.38) and periodic boundary conditions on a mesh of 100 grid cells. Computed with the Godunov flux using the minmod, superbee and MC limiters. `[waveDisc_2nd.m]`

7.7. Numerical experiments

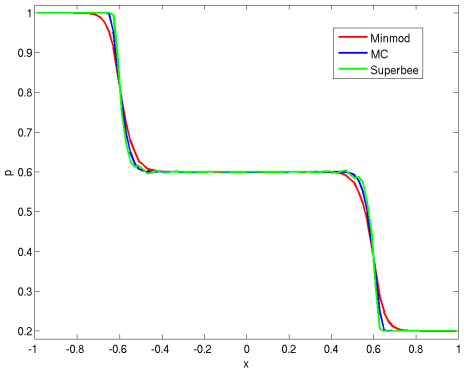
In this section we repeat the numerical experiments carried out in Section 7.5 using second order numerical scheme based on slope limiters. The computational results for the numerical experiment 1 are presented in Figure 7.4, where we compare the performance of the minmod, superbee and MC limiters. We note that both MC and superbee produce some oscillations. On the other hand, minmod is more diffusive compared to the other limiters. Similar conclusion can be drawn from the computational results of numerical experiment 2, presented in Figure 7.5.

In Figure 7.6 we present computational results for the linearized Euler equation using data from numerical experiment 3. Here, the solutions are much sharper than that of the first-order scheme. Similar to numerical experiments 1 and 2, we see that superbee and MC can produce some oscillation, whereas minmod is slightly more diffusive. Note in particular that the stationary shock in density is perfectly resolved, as for the first-order Godunov scheme.



(a) ρ

(b) u



(c) p

FIGURE 7.6. Linearized Euler equation (7.10) computed up to time $t = 0.5$ with initial conditions (7.39) and outflow boundary conditions. Computed with the Godunov flux using the minmod, superbee and MC limiters. `[linEuler_2nd.m]`

CHAPTER 8

Nonlinear hyperbolic systems in one space dimension

A large portion of the partial differential equations of the form

$$\begin{aligned} \partial_t \mathbf{U} + \partial_x \mathbf{f}(\mathbf{U}) &= 0 \\ \mathbf{U}(x,0) &= \mathbf{U}_0(x) \end{aligned} \tag{8.1}$$

which appear in the physical sciences are neither scalar equations nor linear systems, but *nonlinear systems* of conservation laws. The most prominent example of a nonlinear system of conservation laws is the Euler equations for compressible gas dynamics which we saw in Section 1.1.3, but there are several other examples, modeling phenomena such as plasma physics, water waves and elastic materials. In this section we study some of the main structural properties of these types of equations and give a tour of what is known about the well-posedness of such equations. In particular we will study the Riemann problem

$$\begin{aligned} \partial_t \mathbf{U} + \partial_x \mathbf{f}(\mathbf{U}) &= 0 \\ \mathbf{U}(x,0) &= \begin{cases} \mathbf{U}_L & \text{if } x < 0 \\ \mathbf{U}_R & \text{if } x > 0, \end{cases} \end{aligned} \tag{8.2}$$

with the aim of determining the general form of the entropy solution. Just as for scalar equations and linear systems, the solution of the Riemann problem is of great importance in designing finite volume methods for (8.1).

In the remainder of this chapter, the unknown $\mathbf{U}$ will be a vector-valued function $\mathbf{U} : \mathbb{R} \times \mathbb{R}_+ \to \mathcal{U}$ which is assumed to lie in, say, $L^1_{\text{loc}}\big(\mathbb{R} \times \mathbb{R}_+, \mathcal{U}\big)$. Here, $\mathcal{U} \subset \mathbb{R}^m$ is the domain for which (8.1) makes physical sense; for instance, the Euler equations (Section 1.1.3) require ρ and E, the first and last components of $\mathbf{U}$, to be nonnegative. The flux function $f : \mathcal{U} \to \mathbb{R}^m$ is assumed to be as smooth as we like, say, C^3.

Just as for scalar equations, the solutions of (8.1) might develop discontinuities after a finite amount of time, regardless of how smooth $\mathbf{U}_0$ is. Hence, we need to interpret the PDE (8.1) in a weak sense. Multiplying the equation by a test function, integrating over space-time and integrating by parts, we arrive at the following definition.

Definition 8.1. *A function* $\mathbf{U} \in L^1_{\text{loc}}\big(\mathbb{R} \times \mathbb{R}_+, \mathcal{U}\big)$ *is a* weak solution *of* (8.1) *if for every test function* $\varphi \in C_c^\infty(\mathbb{R} \times [0,\infty))$,

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}} \mathbf{U} \partial_t \varphi + \mathbf{f}(\mathbf{U}) \partial_x \varphi \, dx \, dt + \int_{\mathbb{R}} \mathbf{U}_0(x) \varphi(x,0) \, dx = 0.$$

With no extra effort we can prove the *Rankine–Hugoniot condition* for systems of equations: If $\mathbf{U}$ is a piecewise C^1 function with only jump-type discontinuities, then the following are equivalent:

- $\mathbf{U}$ is a weak solution of (8.1)
- $\mathbf{U}$ is a classical solution wherever it is C^1, and satisfies the Rankine–Hugoniot condition

$$\mathbf{f}(\mathbf{U}^+) - \mathbf{f}(\mathbf{U}^-) = s(\mathbf{U}^+ - \mathbf{U}^-) \tag{8.3}$$

across every discontinuity $x = \gamma(t)$. Here, $s = \gamma'(t)$ and $\mathbf{U}^\pm = \lim_{y \to \gamma(t)^\pm} \mathbf{U}(y,t)$.

Note that (8.3) is now a system of m equations but has $2m+1$ unknowns: s and the components of $\mathbf{U}^-$ and $\mathbf{U}^+$. As we will see in Section 8.3, the entropy condition will put constraints on the remaining $m+1$ unknowns.

8.1. Structural properties

Without further structural assumptions, the initial value problem (8.1) is too general to develop a well-posedness theory. In this section we impose several conditions on the flux function f which will enable an existence and uniqueness theory.

Definition 8.2. *The system* (8.1) *is* hyperbolic *if the Jacobian* $\mathbf{f}'(\mathbf{U})$ *is real diagonalizable for every* $\mathbf{U} \in \mathcal{U}$*, that is, there exists an invertible matrix* $R(\mathbf{U}) \in \mathbb{R}^{m\times m}$ *and numbers* $\lambda_1(\mathbf{U}) \leqslant \ldots \leqslant \lambda_m(\mathbf{U}) \in \mathbb{R}$ *such that*

$$\mathbf{f}'(\mathbf{U}) = R(\mathbf{U})\Lambda(\mathbf{U})R(\mathbf{U})^{-1}, \qquad \Lambda(\mathbf{U}) := \operatorname{diag}\big(\lambda_1(\mathbf{U}), \ldots, \lambda_m(\mathbf{U})\big).$$

The system (8.1) *is* strictly hyperbolic *if the eigenvalues are distinct, i.e.* $\lambda_1(\mathbf{U}) < \cdots < \lambda_m(\mathbf{U})$.

Definition 8.3. *Consider a hyperbolic system* (8.1) *and let* $j \in \{1, \ldots, m\}$*. We say that* the jth wave family is genuinely nonlinear *if* $\nabla\lambda_j(\mathbf{U}) \cdot \mathbf{r}_j(\mathbf{U}) \neq 0$ *for all* $\mathbf{U} \in \mathcal{U}$*. We say that* the jth wave family is linearly degenerate *if* $\nabla\lambda_j(\mathbf{U}) \cdot \mathbf{r}_j(\mathbf{U}) = 0$ *for all* $\mathbf{U} \in \mathcal{U}$.

Example 8.4. Consider a scalar conservation law, i.e. (8.1) with $m = 1$. The eigenvalue of the 1×1 matrix $f'(U)$ is just $\lambda_1(U) = f'(U)$, with corresponding eigenvector, say, $r_1(U) = 1$. Since f is assumed to be real-valued, the eigenvalue $\lambda_1(U)$ is always real, so a scalar conservation law is always hyperbolic. Since $\nabla\lambda_1(U) = f''(U)$, we find that a scalar conservation law is genuinely nonlinear if and only if either $f''(U) > 0$ or $f''(U) < 0$ for all $U \in \mathbb{R}$, and it is linearly degenerate if and only if f is linear.

Example 8.5. Consider the linear system of equations (7.2), i.e. the equation (8.1) with $\mathbf{f}(\mathbf{U}) = A\mathbf{U}$ for some constant matrix $A \in \mathbb{R}^{m\times m}$. This system is hyperbolic if A is real diagonalizable (so the concept of hyperbolicity coincides with that of Chapter 7). All of the eigenvalues λ_j are constant with respect to $\mathbf{U}$, so $\nabla\lambda_j \equiv 0$, meaning that *every wave family of a linear system is linearly degenerate.*

Example 8.6. The shallow water equations

$$\begin{aligned} \partial_t h + \partial_x(hv) &= 0 \\ \partial_t(hv) + \partial_x\left(\frac{1}{2}gh^2 + hv^2\right) &= 0 \end{aligned} \tag{8.4}$$

are a model for water waves in a shallow body of water (such as a river, lake or an ocean). The unknowns are $h = h(x,t)$, the water depth at the (horizontal) position x at time t, and $v = v(x,t)$, the horizontal velocity of the column of water at x. The parameter g is the gravitational constant, approximately $9.81 m/s^2$. The conserved variables are h and $m := hv$, the momentum, and we can write (8.4) as the system (8.1) with

$$\mathbf{U} = \begin{pmatrix} h \\ m \end{pmatrix}, \qquad \mathbf{f}(\mathbf{U}) = \begin{pmatrix} m \\ \frac{1}{2}gh^2 + \frac{m^2}{h} \end{pmatrix}.$$

It is easy to compute the eigenvalues and eigenvectors of $\mathbf{f}'(\mathbf{U})$,

$$\lambda_1(\mathbf{U}) = v - c, \quad \lambda_2(\mathbf{U}) = v + c, \quad \mathbf{r}_1(\mathbf{U}) = \begin{pmatrix} 1 \\ v - c \end{pmatrix}, \quad \mathbf{r}_2(\mathbf{U}) = \begin{pmatrix} 1 \\ v + c \end{pmatrix}$$

where $c := \sqrt{gh}$. The eigenvalues are real as long as $h \geqslant 0$, and the matrix $R(\mathbf{U}) = \begin{pmatrix} \mathbf{r}_1(\mathbf{U}) & \mathbf{r}_2(\mathbf{U}) \end{pmatrix}$ is invertible if $h \neq 0$. We conclude that the shallow water equations is strictly hyperbolic in the domain

$$\mathcal{U} = \big\{(h,m) \in \mathbb{R}^2 \,:\, h > 0\big\}.$$

A straightforward computation shows that

$$\nabla\lambda_1(\mathbf{U}) \cdot \mathbf{r}_1(\mathbf{U}) = -\frac{3}{2}\sqrt{\frac{g}{h}}, \qquad \nabla\lambda_2(\mathbf{U}) \cdot \mathbf{r}_2(\mathbf{U}) = \frac{3}{2}\sqrt{\frac{g}{h}},$$

so both wave families are genuinely nonlinear for $\mathbf{U} \in \mathcal{U}$.

Example 8.7. Consider the (one-dimensional) compressible Euler equations for a polytropic gas,

$$\begin{aligned} \partial_t\rho + \partial_x(\rho v) &= 0 \\ \partial_t(\rho v) + \partial_x\big(\rho v^2 + p\big) &= 0 \\ \partial_t E + \partial_x\big((E+p)v\big) &= 0. \end{aligned} \tag{8.5}$$

This is an equation for the motion of a gas in a (one-dimensional) container, and the unknowns ρ, v, p and E are the mass density, velocity, pressure and energy, respectively (see also Section 1.1.3). The pressure and energy are related through the equation of state

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2.$$

The eigenvalues of the Jacobian $\mathbf{f}'(\mathbf{U})$ are

$$\lambda_1(\mathbf{U}) = v - c, \qquad \lambda_2(\mathbf{U}) = v, \qquad \lambda_3(\mathbf{U}) = v + c$$

and the corresponding eigenvectors are

$$\mathbf{r}_1(\mathbf{U}) = \begin{pmatrix} 1 \\ v - c \\ H - vc \end{pmatrix}, \qquad \mathbf{r}_2(\mathbf{U}) = \begin{pmatrix} 1 \\ v \\ \frac{1}{2}v^2 \end{pmatrix}, \qquad \mathbf{r}_3(\mathbf{U}) = \begin{pmatrix} 1 \\ v + c \\ H + vc \end{pmatrix}.$$

Here, $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound and $H = \frac{E+p}{\rho}$ is the *total specific enthalpy*. The eigenvalues are real as long as $\rho > 0$, $p \geqslant 0$, and the matrix of eigenvectors is invertible as long as $\rho, p > 0$. Thus, the Euler equations is hyperbolic in the domain

$$\mathcal{U} = \big\{(\rho, m, E) \in \mathbb{R}^3 \ : \ \rho > 0, \ E > \tfrac{m^2}{2\rho}\big\}.$$

By a straightforward calculation it can be shown that the first and third wave families are genuinely nonlinear, while the second wave family is linearly degenerate.

8.2. Simple solutions

We will henceforth assume that *every* wave family is either genuinely nonlinear or linearly degenerate. Moreover, when the jth wave family is genuinely nonlinear we will normalize the eigenvector $\mathbf{r}_j$ so that

$$\nabla\lambda_j(\mathbf{U}) \cdot \mathbf{r}_j(\mathbf{U}) = 1 \qquad \text{for all } \mathbf{U} \in \mathcal{U}. \tag{8.6}$$

In this section we will look for some particular solutions of the Riemann problem (8.3) called *simple solutions*, with the goal of "gluing" these together to solve the general Riemann problem in Section 8.4. For the remainder of this section we fix $\mathbf{U}_L \in \mathcal{U}$, and we will try to determine the set of all $\mathbf{U}_R \in \mathcal{U}$ such that the resulting Riemann problem (8.3) has a particular type of solution, such as a rarefaction wave or a shock wave.

8.2.1. Rarefaction waves. Recall that a *rarefaction wave* is a smooth solution of (8.2). Based on the observation that the initial value problem (8.2) is invariant under the transformation $(x, t) \mapsto (\alpha x, \alpha t)$ for any $\alpha > 0$ (that is, if $\mathbf{U}$ is a solution then also $\mathbf{U}(\alpha x, \alpha t)$ is a solution), we look for solutions which are invariant to this transformation:

$$\mathbf{U}(x, t) = \mathbf{u}(x/t)$$

for some differentiable $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$. Inserting into (8.1) we find that

$$0 = \partial_t \mathbf{u}(x/t) + \partial_x \mathbf{f}(\mathbf{u}(x/t)) = -\frac{x}{t^2}\mathbf{u}'(x/t) + \frac{1}{t}\mathbf{f}'(\mathbf{u}(x/t))\mathbf{u}'(x/t),$$

or written in terms of $\xi := x/t$,

$$\mathbf{f}'(\mathbf{u}(\xi))\mathbf{u}'(\xi) = \xi\mathbf{u}'(\xi). \tag{8.7}$$

This can only mean one of two things: Either $\mathbf{u}'(\xi) = 0$, or $\mathbf{u}'(\xi)$ is an eigenvector of the Jacobian $\mathbf{f}'(\mathbf{u}(\xi))$ with corresponding eigenvalue ξ. In the latter case we can write

$$\mathbf{u}'(\xi) = \mathbf{r}_j(\mathbf{u}(\xi)), \qquad \xi = \lambda_j(\mathbf{u}(\xi)) \tag{8.8}$$

for some $j \in \{1, \dots, m\}$ (at least up to a scalar multiple of $\mathbf{r}_j$). From the second identity in (8.8) we see that if $\mathbf{u}(\xi_L) = \mathbf{U}_L$ and $\mathbf{u}(\xi_R) = \mathbf{U}_R$ for some $\xi_L, \xi_R \in \mathbb{R}$, then $\xi_L = \lambda_j(\mathbf{U}_L)$ and $\xi_R = \lambda_j(\mathbf{U}_R)$. Therefore, the solution will be the function

$$\mathbf{U}(x, t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < \lambda_j(\mathbf{U}_L) \\ \mathbf{u}(\frac{x}{t}) & \lambda_j(\mathbf{U}_L) < \frac{x}{t} < \lambda_j(\mathbf{U}_R) \\ \mathbf{U}_R & \lambda_j(\mathbf{U}_R) < \frac{x}{t}. \end{cases} \tag{8.9}$$

If we differentiate the second identity in (8.8) with respect to ξ we obtain

$$1 = \partial_\xi \lambda_j(\mathbf{u}(\xi)) = \nabla\lambda_j(\mathbf{u}(\xi)) \cdot \mathbf{u}'(\xi) = \nabla\lambda_j(\mathbf{u}(\xi)) \cdot \mathbf{r}_j(\mathbf{u}(\xi)),$$

where we have first used the chain rule and then the first identity in (8.8). Note that the particular normalization $\nabla\lambda_j \cdot \mathbf{r}_j = 1$ is the normalization which we imposed on genuinely nonlinear wave families in (8.6).

The above computations motivate the following construction of a *family* of solutions of the Riemann problem. Let $\mathbf{W}_j = \mathbf{W}_j(\varepsilon)$ be the solution of the ODE

$$\mathbf{W}_j'(\varepsilon) = \mathbf{r}_j(\mathbf{W}(\varepsilon)), \qquad \mathbf{W}_j(0) = \mathbf{U}_L \tag{8.10}$$

(which is merely a reparametrization of the first identity in (8.8) with $\varepsilon = \xi - \lambda_j(\mathbf{U}_L)$). In other words, $\mathbf{W}_j$ is a parametrization of the integral curve of the vector field $\mathbf{r}_j$ that goes through $\mathbf{U}_L$. By the standard theory of ODEs, the problem (8.10) has a unique solution for $\varepsilon \in (-\bar\varepsilon, \bar\varepsilon)$ for some $\bar\varepsilon > 0$. If $\mathbf{U}_R$ lies anywhere on the integral curve $\mathcal{R}_j(\mathbf{U}_L) = \{\mathbf{W}_j(\varepsilon) \ : \ 0 \leqslant \varepsilon < \bar\varepsilon\}$ then the Riemann problem (8.3) has the solution (8.9) with $\mathbf{u}(\xi) = \mathbf{W}_j\big(\xi - \lambda_j(\mathbf{U}_L)\big)$. (Here we exclude values $\varepsilon < 0$, for if ε were negative then $\lambda_j(\mathbf{U}_R) = \varepsilon + \lambda_j(\mathbf{U}_L) < \lambda_j(\mathbf{U}_L)$, in which case the formula (8.9) would not make sense.)

Lemma 8.8. *Let the jth wave family be genuinely nonlinear and let $\mathbf{U}_L \in \mathcal{U}$. Then there is a curve*

$$\mathcal{R}_j(\mathbf{U}_L) = \{\mathbf{W}_j(\varepsilon) \ : \ 0 \leqslant \varepsilon < \bar\varepsilon\} \tag{8.11}$$

emanating from $\mathbf{U}_L$ such that if $\mathbf{U}_R \in \mathcal{R}_j(\mathbf{U}_L)$ then

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < \lambda_j(\mathbf{U}_L) \\ \mathbf{W}_j\big(\frac{x}{t} - \lambda_j(\mathbf{U}_L)\big) & \lambda_j(\mathbf{U}_L) < \frac{x}{t} < \lambda_j(\mathbf{U}_R) \\ \mathbf{U}_R & \lambda_j(\mathbf{U}_L) < \frac{x}{t}. \end{cases} \tag{8.12}$$

solves the Riemann problem (8.2).

8.2.2. Contact discontinuities. As we found in the previous section, a linearly degenerate wave family cannot have rarefaction wave solutions, so we should expect discontinuous solutions whenever the jth wave family is linearly degenerate. It is still natural to ask what the integral curve (8.10) yields in this case. Differentiating $\lambda_j(\mathbf{W}_j(\varepsilon))$ with respect to ε now yields

$$\frac{d}{d\varepsilon}\lambda_j(\mathbf{W}_j(\varepsilon)) = \nabla\lambda_j(\mathbf{W}_j(\varepsilon)) \cdot \mathbf{r}_j(\mathbf{W}_j(\varepsilon)) \equiv 0,$$

so it follows that $\lambda_j(\mathbf{W}_j(\varepsilon)) \equiv \lambda_j(\mathbf{U}_L)$ for every $\varepsilon \in (-\bar\varepsilon, \bar\varepsilon)$. We then find that

$$\begin{aligned} \frac{d}{d\varepsilon}\big(\mathbf{f}(\mathbf{W}_j) - \lambda_j(\mathbf{W}_j)\mathbf{W}_j\big) &= \mathbf{f}'(\mathbf{W}_j)\frac{d\mathbf{W}_j}{d\varepsilon} - \lambda_j(\mathbf{W}_j)\frac{d\mathbf{W}_j}{d\varepsilon} \\ &= \mathbf{f}'(\mathbf{W}_j)\mathbf{r}_j - \lambda_j(\mathbf{W}_j)\mathbf{r}_j \\ &= \lambda_j(\mathbf{W}_j)\mathbf{r}_j - \lambda_j(\mathbf{W}_j)\mathbf{r}_j \\ &= 0. \end{aligned}$$

Hence, if $\mathbf{U}_R = \mathbf{W}_j(\varepsilon)$ for any $\varepsilon \in (-\bar\varepsilon, \bar\varepsilon)$, then $\mathbf{f}(\mathbf{U}_R) - \lambda_j(\mathbf{U}_R)\mathbf{U}_R = \mathbf{f}(\mathbf{U}_L) - \lambda_j(\mathbf{U}_L)\mathbf{U}_L$, which can be rewritten as

$$\mathbf{f}(\mathbf{U}_R) - \mathbf{f}(\mathbf{U}_L) = s\big(\mathbf{U}_R - \mathbf{U}_L\big), \qquad s := \lambda_j(\mathbf{U}_L) = \lambda_j(\mathbf{U}_R).$$

This is precisely the Rankine–Hugoniot condition for a discontinuity moving with speed s. We conclude:

Lemma 8.9. *Let the jth wave family be linearly degenerate and let $\mathbf{U}_L \in \mathcal{U}$. Then there is a curve*

$$\mathcal{C}_j(\mathbf{U}_L) = \big\{\mathbf{W}_j(\varepsilon) \ : \ \varepsilon \in (-\bar\varepsilon, \bar\varepsilon)\big\} \tag{8.13}$$

passing through $\mathbf{U}_L$ such that if $\mathbf{U}_R \in \mathcal{C}_j(\mathbf{U}_L)$ then

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < \lambda_j(\mathbf{U}_L) \\ \mathbf{U}_R & \lambda_j(\mathbf{U}_L) < \frac{x}{t} \end{cases} \tag{8.14}$$

is a weak solution of the Riemann problem (8.2).

A solution such as the above, where λ_j remains constant across a jump along the jth eigenvector, is called a *contact discontinuity*. Contact discontinuities appear in gas dynamics when a discontinuity in the mass density (but not in the pressure or velocity) is transported along with the gas. This is to be distinguished from *shock waves*, which move faster than the gas itself, and are characterized by a discontinuous increase in pressure.

8.2.3. The Hugoniot locus. We have found a class of discontinuous simple solutions corresponding to linearly degenerate wave families, as well as smooth simple solutions corresponding to genuinely nonlinear wave families. It remains to determine the discontinuous simple solutions corresponding to genuinely nonlinear wave families.

Fix some $\mathbf{U}_L \in \mathcal{U}$. Since all discontinuous solutions must satisfy the Rankine–Hugoniot condition, we define

$$\mathcal{H}(\mathbf{U}_L) = \big\{\mathbf{U}_R \in \mathcal{U} \ : \ \exists\, s \in \mathbb{R} \text{ such that } \mathbf{f}(\mathbf{U}_R) - \mathbf{f}(\mathbf{U}_L) = s\big(\mathbf{U}_R - \mathbf{U}_L\big)\big\}. \tag{8.15}$$

This set is called the *Hugoniot locus*, and consist of all $\mathbf{U}_R \in \mathcal{U}$ such that

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < s \\ \mathbf{U}_R & s < \frac{x}{t} \end{cases} \tag{8.16}$$

(for some $s \in \mathbb{R}$) is a weak solution of the Riemann problem (8.2). Clearly, the contact discontinuity curve $\mathcal{C}_j(\mathbf{U}_L)$ is a subset of the Hugoniot locus.

Lemma 8.10. *Assume that* (8.1) *is strictly hyperbolic and let* $\mathbf{U}_L \in \mathcal{U}$. *Then there exist* m *curves* $\mathcal{H}_1(\mathbf{U}_L), \dots, \mathcal{H}_m(\mathbf{U}_L)$ *passing through* $\mathbf{U}_L$ *such that*

$$\mathcal{H}(\mathbf{U}_L) = \mathcal{H}_1(\mathbf{U}_L) \cup \cdots \cup \mathcal{H}_m(\mathbf{U}_L).$$

Moreover, each curve $\mathcal{H}_j(\mathbf{U}_L)$ *can be parametrized by some function* $\mathbf{W}_j(\varepsilon)$ *satisfying*

$$\mathbf{W}_j(0) = \mathbf{U}_L, \qquad \mathbf{W}_j'(0) = \mathbf{r}_j(\mathbf{U}_L). \tag{8.17}$$

Sketch of proof. The idea is to use the fundamental theorem of calculus to write $\mathbf{f}(\mathbf{U}_R) - \mathbf{f}(\mathbf{U}_L) = M(\mathbf{U}_L, \mathbf{U}_R)\big(\mathbf{U}_R - \mathbf{U}_L\big)$, where $M(\mathbf{U}_L, \mathbf{U}_R) = \int_0^1 \mathbf{f}'\big(\mathbf{U}_L + \tau(\mathbf{U}_R - \mathbf{U}_L)\big)\, d\tau$. We can then write

$$\mathcal{H}(\mathbf{U}_L) = \big\{\mathbf{U}_R \in \mathcal{U} \ : \ \exists\, s \in \mathbb{R} \text{ such that } M(\mathbf{U}_L, \mathbf{U}_R)\big(\mathbf{U}_R - \mathbf{U}_L\big) = s\big(\mathbf{U}_R - \mathbf{U}_L\big)\big\},$$

in other words, $\mathbf{U}_R - \mathbf{U}_L$ must be an eigenvector of $M(\mathbf{U}_L, \mathbf{U}_R)$. The matrix $M(\mathbf{U}_L, \mathbf{U}_L) = \mathbf{f}'(\mathbf{U}_L)$ is real diagonalizable, so for $\mathbf{U}_R$ within some $\bar{\varepsilon}$-distance of $\mathbf{U}_L$, the matrix $M(\mathbf{U}_L, \mathbf{U}_R)$ is still real diagonalizable. An application of the Implicit Function Theorem now yields the existence of m distinct curves $\mathbf{W}_1(\varepsilon), \dots, \mathbf{W}_m(\varepsilon)$ such that $\mathbf{W}_j(0) = \mathbf{U}_L$ and

$$M\big(\mathbf{U}_L, \mathbf{W}_j(\varepsilon)\big)\big(\mathbf{W}_j(\varepsilon) - \mathbf{U}_L\big) = s\big(\mathbf{W}_j(\varepsilon) - \mathbf{U}_L\big).$$

Dividing by ε and passing $\varepsilon \to 0$, we see that $\mathbf{W}_j'(0)$ is an eigenvector of $M(\mathbf{U}_L, \mathbf{U}_L) = \mathbf{f}'(\mathbf{U}_L)$. After possibly reordering and reparametrizing $\mathbf{W}_1, \dots, \mathbf{W}_m$ we can conclude that $\mathbf{W}_j'(0) = \mathbf{r}_j(\mathbf{U}_L)$. □

8.3. Entropy conditions

Given any left-hand state $\mathbf{U}_L$, the Hugoniot locus provides an entire family of right-hand states $\mathbf{U}_R$ for which the Riemann problem has a discontinuous solution. However, just as for scalar conservation laws, we should expect some of these solutions to be physically unrealistic, and we need to impose further *entropy conditions* to single out a unique, physically relevant solution. To this end, consider the following *parabolic regularization* of (8.1):

$$\partial_t \mathbf{U} + \partial_x \mathbf{f}(\mathbf{U}) = \nu \partial_{xx} \mathbf{U} \tag{8.18}$$

for some viscosity parameter $\nu > 0$. We would like to consider only those weak solutions of (8.2) which arise as the limit $\nu \to 0$ of the regularized equation (8.18). As for scalar conservation laws, we study the limit $\nu \to 0$ in terms of the *entropy* of the solution. To this end, let $\eta : \mathcal{U} \to \mathbb{R}$ be a convex function and take the inner product of (8.18) with $\eta'(\mathbf{U})$:

$$\eta'(\mathbf{U})^T \partial_t \mathbf{U} + \eta(\mathbf{U})^T f'(\mathbf{U}) \partial_x \mathbf{U} = \nu \eta'(\mathbf{U})^T \partial_{xx} \mathbf{U},$$

where ^{T} denotes the transpose. By manipulations similar to those in Section 3.3, we can write the above as

$$\partial_t \eta(\mathbf{U}) + \eta(\mathbf{U})^T f'(\mathbf{U}) \partial_x \mathbf{U} = \nu \partial_{xx} \eta(\mathbf{U}) - \nu (\partial_x \mathbf{U})^T \eta''(\mathbf{U})(\partial_x \mathbf{U}). \tag{8.19}$$

Here, η'' is the Hessian of η. Assume now that there is a function $q : \mathcal{U} \to \mathbb{R}$ such that $q'(\mathbf{U})^T = \eta(\mathbf{U})^T \mathbf{f}'(\mathbf{U})$ for all $\mathbf{U} \in \mathcal{U}$. We can then write $\eta(\mathbf{U})^T f'(\mathbf{U}) \partial_x \mathbf{U} = q'(\mathbf{U})^T \partial_x \mathbf{U} = \partial_x q(\mathbf{U})$.

Definition 8.11. *A pair of functions $\eta : \mathcal{U} \to \mathbb{R}$ and $q : \mathcal{U} \to \mathbb{R}$ is an* entropy pair *for* (8.1) *if η is strictly convex and q satisfies $q'(\mathbf{U})^T = \eta(\mathbf{U})^T \mathbf{f}'(\mathbf{U})$ for all $\mathbf{U} \in \mathcal{U}$.*

By the assumption of convexity, the Hessian η'' is positive definite, and so the second term on the right-hand side of (8.19) is nonpositive. In the limit $\nu \to 0$ we therefore obtain

$$\partial_t \eta(\mathbf{U}) + \partial_x q(\mathbf{U}) \leqslant 0. \tag{8.20}$$

This will be our entropy condition.

Definition 8.12. *A weak solution $\mathbf{U}$ of* (8.1) *is an* entropy solution *if for all entropy pairs (η, q) the entropy inequality* (8.20) *holds in the sense of distributions, that is,*

$$\int_{\mathbb{R}_+} \int_{\mathbb{R}} \eta(\mathbf{U}) \partial_t \varphi + q(\mathbf{U}) \partial_x \varphi \, dx \, dt + \int_{\mathbb{R}} \eta(\mathbf{U}_0(x)) \varphi(x, 0) \, dx \geqslant 0 \qquad \forall\ \varphi \in C_c^\infty(\mathbb{R} \times \mathbb{R}_+).$$

Example 8.13. Consider the shallow water equations, Example 8.6. The *energy* of a state $\mathbf{U} \in \mathcal{U}$ is defined as

$$\eta(\mathbf{U}) = \frac{1}{2} g h^2 + \frac{1}{2} h v^2,$$

which is the sum of potential and kinetic energy. Writing η in terms of conserved variables, $\eta(\mathbf{U}) = \frac{1}{2} g h^2 + \frac{m^2}{2h}$, it is clear that η is strictly convex for $h > 0$. If we define $q(\mathbf{U}) = \frac{1}{2} h v^3 + h^2 v$ then it is straightforward to show that (η, q) is an entropy pair for the shallow water equations.

Example 8.14. Consider the compressible Euler equations for a polytropic gas, Example 8.7. The *thermodynamic entropy* is defined as

$$\eta(\mathbf{U}) = -\frac{\rho s}{\gamma - 1}$$

where $s = \log\left(\frac{p}{\rho^\gamma}\right)$ is the *specific entropy*. If we define $q(\mathbf{U}) = -\frac{\rho v s}{\gamma - 1}$ then (η, q) is an entropy pair for (8.5).

Remark 8.15. *In most nonlinear hyperbolic systems which appear in the physical sciences, there is only one entropy pair available. This is in stark contrast to scalar conservation law, where* any *convex function η gives rise to an entropy pair.*

In the particular case where $\mathbf{U}$ is a piecewise C^1 function with jump discontinuities across smooth curves, we can simplify the entropy condition in Definition 8.12 greatly, just as for scalar conservation laws (cf. Theorem 3.8). If $\mathbf{U}^-$, $\mathbf{U}^+$ denote the values to the left and to the right of a jump discontinuity with speed s, then the following are equivalent:

- $\mathbf{U}$ is an entropy solution of (8.1)
- $\mathbf{U}$ is a classical solution of (8.1) wherever it is C^1, and at jump discontinuities it satisfies

$$\big(q(\mathbf{U}^+) - q(\mathbf{U}^-)\big) - s\big(\eta(\mathbf{U}^+) - \eta(\mathbf{U}^-)\big) \leqslant 0 \tag{8.21}$$

  for every entropy pair (η, q).

The proof of the above equivalence is more or less the same as for scalar equations, and follows by the same approach as the proof of the Rankine–Hugoniot condition.

We now determine which parts of the Hugoniot locus $\mathcal{H}(\mathbf{U}_L) = \mathcal{H}_1(\mathbf{U}_L) \cup \cdots \cup \mathcal{H}_m(\mathbf{U}_L)$ satisfies the entropy condition (8.21). For each $j \in \{1, \ldots, m\}$, we must consider two cases:

The jth wave family is linearly degenerate: Here we know that $\mathcal{H}_j(\mathbf{U}_L) = \mathcal{C}_j(\mathbf{U}_L)$, and that this curve is parametrized by the integral curve (8.10). Define the quantity

$$E(\varepsilon) = \big(q(\mathbf{W}_j(\varepsilon)) - q(\mathbf{U}_L)\big) - s(\mathbf{W}_j(\varepsilon))\big(\eta(\mathbf{W}_j(\varepsilon)) - \eta(\mathbf{U}_L)\big).$$

From (8.21) we see that a discontinuity with left- and right-hand sides $\mathbf{U}_L$ and $\mathbf{U}_R := \mathbf{W}_j(\varepsilon)$ satisfies the entropy condition *if and only if* $E(\varepsilon) \leqslant 0$. Writing $\mathbf{W} = \mathbf{W}_j(\varepsilon)$ and $s = s(\mathbf{W}_j(\varepsilon))$, we have

$$\begin{aligned} E'(\varepsilon) &= \eta'(\mathbf{W}) \cdot f'(\mathbf{W})\mathbf{W}' - s'(\eta(\mathbf{W}) - \eta(\mathbf{U}_L)) - s\eta'(\mathbf{W}) \cdot \mathbf{W}' \\ &= \eta'(\mathbf{W}) \cdot \big(f'(\mathbf{W})\mathbf{W}' - s\mathbf{W}'\big) - s'(\eta(\mathbf{W}) - \eta(\mathbf{U}_L)) \\ &= 0, \end{aligned}$$

where we have used the fact that $\mathbf{W}'$ is an eigenvector of f' with eigenvalue $s \equiv \lambda_j(\mathbf{U}_L)$. Since $E(0) = 0$ we can conclude that contact discontinuities always satisfy the entropy condition.

The jth wave family is genuinely nonlinear: In this case the computation is somewhat more involved. Let $\mathbf{W}_j(\varepsilon)$ be the parametrization of the curve $\mathcal{H}_j(\mathbf{U}_L)$ given in Lemma 8.10, and let $s(\varepsilon)$ be the speed of the corresponding discontinuity. By differentiating the relation $\mathbf{f}(\mathbf{W}_j(\varepsilon)) - \mathbf{f}(\mathbf{U}_L) = s(\varepsilon)\big(\mathbf{W}_j(\varepsilon) - \mathbf{U}_L\big)$ twice, it is straightforward to show that

$$\lambda_j(0) = s(0), \qquad \lambda_j'(0) = 2s'(0) = 1 \tag{8.22}$$

(where $\lambda_j(\varepsilon) = \lambda(\mathbf{W}_j(\varepsilon))$). A long and tedious computation shows that $E'(0) = E''(0) = 0$, while

$$E'''(0) = \frac{1}{2}\mathbf{r}^T\eta''(\mathbf{U}_L)\mathbf{r}, \qquad \mathbf{r} = \mathbf{r}_j(\mathbf{U}_L).$$

The Hessian matrix η'' is positive definite since η is strictly convex, so for ε small, we find that $E(\varepsilon) < 0$ *if and only if* $\varepsilon < 0$. From (8.22) we see that, up to terms of order ε^2,

$$\lambda_j(\varepsilon) = \lambda_j(\mathbf{U}_L) + \varepsilon, \qquad s(\varepsilon) = \lambda_j(\mathbf{U}_L) + \frac{\varepsilon}{2}.$$

Solving for ε, we can state the condition $\varepsilon < 0$ equivalently in terms of the shock speed s and the eigenvalues:

$$\lambda_j(\mathbf{U}_R) < s < \lambda_j(\mathbf{U}_L) \tag{8.23a}$$

If the system is strictly hyperbolic, i.e. $\lambda_1(\mathbf{U}) < \cdots < \lambda_m(\mathbf{U})$, then we can also deduce that (at least for ε small)

$$\lambda_{j-1}(\mathbf{U}_L) < s < \lambda_{j+1}(\mathbf{U}_R) \tag{8.23b}$$

The condition (8.23) is the *Lax entropy condition*, and it can be interpreted geometrically as follows: The characteristics in the jth wave family (i.e., curves in the x-t-plane with slope $\lambda_j(\mathbf{U})$) *impinge* on the shock, whereas the characteristics in all other wave families *go through* the shock. Counting characteristic curves, we see that (8.23) asserts that $m - j + 1$ characteristics impinge on the shock from the left, while j characteristics impinge on it from the right—a total of $m + 1$ conditions. Taken together with the m equations in the Rankine–Hugoniot condition (8.3), we see that the Lax entropy condition provides enough information to determine the $2m + 1$ unknowns s and $\mathbf{U}_L$, $\mathbf{U}_R$.

Lemma 8.16. *Assume that* (8.1) *is strictly hyperbolic with only linearly degenerate or genuinely nonlinear wave families. Consider a piecewise C^1 weak solution* $\mathbf{U}$ *of* (8.1) *with sufficiently small jump discontinuities. Then* $\mathbf{U}$ *is an entropy solution of* (8.1) if and only if *at every jump discontinuity there is an index $j \in \{1, \dots, m\}$ such that either:*

- *the jth wave family is linearly degenerate, or*
- *the jth wave family is genuinely nonlinear, and the Lax entropy condition* (8.23) *holds.*

We have also found that for every genuinely nonlinear wave family j, if we restrict the Hugoniot curve $\mathcal{H}_j(\mathbf{U}_L) = \{\mathbf{W}_j(\varepsilon) : \varepsilon \in (-\bar{\varepsilon}, \bar{\varepsilon})\}$ to values $\varepsilon < 0$, then the discontinuous solution (8.16) is an entropy solution.

Lemma 8.17. *Let the jth wave family be genuinely nonlinear and let $\mathbf{U}_L \in \mathcal{U}$. Then there is a curve*

$$\mathcal{S}_j(\mathbf{U}_L) = \big\{\mathbf{W}_j(\varepsilon) \ : \ -\bar{\varepsilon} < \varepsilon \leqslant 0\big\} \tag{8.24}$$

emanating from $\mathbf{U}_L$ such that if $\mathbf{U}_R \in \mathcal{S}_j(\mathbf{U}_L)$ then there is some $s \in \mathbb{R}$ such that

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < s \\ \mathbf{U}_R & s < \frac{x}{t} \end{cases} \tag{8.25}$$

is an entropy solution of the Riemann problem (8.2).

8.4. The Riemann problem

To solve the general Riemann problem (8.2) we will assume that the system is strictly hyperbolic and that each wave family is either genuinely nonlinear or linearly degenerate. In the previous sections we have shown that through every $\mathbf{U}_L \in \mathcal{U}$, there are m curves $\mathcal{W}_1(\mathbf{U}_L), \dots, \mathcal{W}_m(\mathbf{U}_L)$ such that if $\mathbf{U}_R$ lies on any of these curves, then the resulting Riemann problem can be solved with a single simple solution—either a rarefaction wave, a contact discontinuity or an (entropy satisfying) shock. To be more precise, we can write the set of all states $\mathbf{U}_R$ which can be connected to $\mathbf{U}_L$ by a simple solution as

$$\mathcal{W}(\mathbf{U}_L) = \mathcal{W}_1(\mathbf{U}_L) \cup \cdots \cup \mathcal{W}_m(\mathbf{U}_L),$$

$$\mathcal{W}_j(\mathbf{U}_L) = \begin{cases} \mathcal{C}_j(\mathbf{U}_L) & \text{if the } j\text{th wave family is linearly degenerate} \\ \mathcal{S}_j(\mathbf{U}_L) \cup \mathcal{R}_j(\mathbf{U}_L) & \text{if the } j\text{th wave family is genuinely nonlinear.} \end{cases} \tag{8.26}$$

Each of the curves $\mathcal{C}_j$, $\mathcal{S}_j$ and $\mathcal{R}_j$ can be parametrized by some function $\mathbf{W}_j(\mathbf{U}_L, \varepsilon)$ for $\varepsilon \in (-\bar{\varepsilon}, \bar{\varepsilon})$, $\varepsilon \in (-\bar{\varepsilon}, 0]$ and $\varepsilon \in [0, \bar{\varepsilon})$, respectively, where $\bar{\varepsilon} > 0$ is some number depending only on $\mathbf{U}_L$. For any $\mathbf{U}_R = \mathbf{W}_j(\mathbf{U}_L, \varepsilon) \in \mathcal{W}(\mathbf{U}_L)$, denote the simple solution of the corresponding Riemann problem by $\mathbf{u}_j(\mathbf{U}_L, \varepsilon; x, t)$—that is, $\mathbf{u}_j(\mathbf{U}_L, \varepsilon; \cdot, \cdot)$ is either of the formulas (8.12), (8.14) or (8.25), depending on whether $\mathbf{U}_R$ lies on $\mathcal{R}_j$, $\mathcal{C}_j$ or $\mathcal{S}_j$, respectively.

To construct the solution of the Riemann problem for general $\mathbf{U}_L, \mathbf{U}_R \in \mathcal{U}$, the idea is to "walk" along each of the wave curves $\mathcal{W}_j$ and "paste" together the corresponding simple solutions, as follows: The state $\mathbf{U}_L$ can be connected to any intermediate state $\mathbf{U}_1 = \mathbf{W}_1(\mathbf{U}_L, \varepsilon_1)$ by the simple solution $\mathbf{u}_1\big(\mathbf{U}_L, \varepsilon_1; x, t\big)$. The state $\mathbf{U}_1$ can again be connected to another state $\mathbf{U}_2 = \mathbf{W}_2(\mathbf{U}_1, \varepsilon_2)$ by the simple solution $\mathbf{u}_2\big(\mathbf{U}_1, \varepsilon_2; x, t\big)$. By carefully choosing $\varepsilon_1, \dots, \varepsilon_m$, the hope is that we finally end up at $\mathbf{U}_m = \mathbf{W}_m(\mathbf{U}_{m-1}, \varepsilon_m) = \mathbf{U}_R$—in other words,

$$\mathbf{U}_R = \mathbf{W}_m\big(\mathbf{W}_{m-1}\big(\dots \mathbf{W}_1\big(\mathbf{U}_L, \varepsilon_1\big), \dots\big), \varepsilon_{m-1}\big), \varepsilon_m\big). \tag{8.27}$$

It is rather straightforward to see that we can indeed find $\varepsilon_1, \dots, \varepsilon_m$ so that we can connect $\mathbf{U}_L$ and $\mathbf{U}_R$ by m waves, as described above. For any $\mathbf{U} \in \mathcal{U}$, each of the curves $\mathcal{W}_1(\mathbf{U}), \dots, \mathcal{W}_m(\mathbf{U})$ is parametrized by some function $\mathbf{W}_j(\mathbf{U}, \varepsilon)$ satisfying $\frac{\partial \mathbf{W}_j}{\partial \varepsilon}(\mathbf{U}, 0) = \mathbf{r}_j(\mathbf{U})$. Since $\mathbf{r}_1(\mathbf{U}), \dots, \mathbf{r}_m(\mathbf{U})$ are linearly independent, and since the curves $\mathcal{W}_1, \dots, \mathcal{W}_m$ are smooth, the collection of curves $\mathcal{W}(\mathbf{U})$ constitute a local coordinate system around $\mathbf{U}$. By an application of the Implicit Function Theorem we deduce that for any $\mathbf{U}_R$ in some open ball around $\mathbf{U}_L$, we can indeed find $\varepsilon_1, \dots, \varepsilon_m$ such that (8.27) holds.

We summarize our conclusions in the following theorem, which was originally published in [**Lax57**].

Theorem 8.18. *Consider a strictly hyperbolic system of conservation laws where every wave family is either genuinely nonlinear or linearly degenerate. Fix some $\mathbf{U}_L \in \mathcal{U}$. Then there is a $\delta > 0$ such that if $|\mathbf{U}_R - \mathbf{U}_L| < \delta$ then the Riemann problem* (8.2) *has a unique solution consisting of $m+1$ constant states $\mathbf{U}_L = \mathbf{U}_0, \mathbf{U}_1, \dots, \mathbf{U}_m = \mathbf{U}_R$ separated by simple solutions:*

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \frac{x}{t} < \sigma_1^- \\ \mathbf{u}_1\big(\mathbf{U}_L, \varepsilon_1; x, t\big) & \sigma_1^- < \frac{x}{t} < \sigma_1^+ \\ \mathbf{U}_1 & \sigma_1^+ < \frac{x}{t} < \sigma_2^- \\ \vdots & \\ \mathbf{U}_{m-1} & \sigma_{m-1}^+ < \frac{x}{t} < \sigma_m^- \\ \mathbf{u}_m\big(\mathbf{U}_{m-1}, \varepsilon_m; x, t\big) & \sigma_m^- < \frac{x}{t} < \sigma_m^+ \\ \mathbf{U}_R & \sigma_m^+ < \frac{x}{t} \end{cases} \tag{8.28}$$

for some $\sigma_1^- \leqslant \sigma_1^+ \leqslant \ldots \leqslant \sigma_m^+$ and $\varepsilon_1, \dots, \varepsilon_m \in \mathbb{R}$.

APPENDIX A

Results from real analysis

Theorem A.1 (Gronwall's inequality)**.** *Let $\beta(t)$ be continuous and $u(t)$ be differentiable on some interval $[a, b]$, and assume that*

$$u'(t) \leqslant \beta(t)u(t) \qquad \forall\ t \in (a, b).$$

Then

$$u(t) \leqslant u(a) \exp\left(\int_a^t \beta(t)\right) \qquad \forall\ t \in [a, b].$$

Bibliography

[CM80] M. G. Crandall and A. Majda. Monotone difference approximations for scalar conservation laws. *Math. Comp.*, 34:121, 1980.

[DS88] N. Dunford and J. T. Schwartz. *Linear operators. Part I*. Wiley Classics Library. John Wiley & Sons Inc., New York, 1988.

[Giu84] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*, volume 80 of *Monographs in Mathematics*. Birkhuser Basel, 1984.

[God59] S. K. Godunov. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations. *Math. Sbornik*, 47:271306, 1959.

[GR91] E. Godlewski and P.A. Raviart. *Hyperbolic Systems of Conservation Laws*. Ellipses, 1991.

[GST01] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong Stability-Preserving High-Order Time Discretization Methods. *SIAM Review*, 43(1):89112, 2001.

[Har83] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49(3):357393, 1983.

[HNW87] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations*. 1987.

[HOEC86] A. Harten, S. Osher, B. Engquist, and S. R. Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Appl. Numer. Math.*, 2(3-5):347378, 1986.

[Hop69] E. Hopf. On the right weak solution of the Cauchy problem for quasilinear equations of first order. *Journal of Mathematics and Mechanics*, 19:483487, 1969.

[HR15] H. Holden and N. H. Risebro. *Front Tracking for Hyperbolic Conservation Laws*. Springer-Verlag Berlin Heidelberg, second edition edition, 2015.

[Kru70] S. N. Kruzkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217243, 1970.

[Lax57] P. D. Lax. Hyperbolic systems of conservation laws II. *Communications on Pure and Applied Mathematics*, 10(4):537566, 1957.

[LL87] L. D. Landau and E. M. Lifschitz. *Fluid Mechanics*. Butterworth-Heinemann, 1987.

[LOC94] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115(1):200212, 1994.

[Ole59] O. A. Oleinik. Uniqueness and stability of the generalized solution of the Cauchy problem for a quasi-linear equation. *Uspekhi Mat. Nauk*, 14(2(86)):165170, 1959. English translation, Amer. Math. Soc. Trans., ser. 2, no. 33, pp. 285-290.

[Shu97] C.-W. Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. Technical report, 1997.

[TW09] A. Tveito and R. Winther. *Introduction to Partial Differential Equations; A Computational Approach*, volume 29. Springer-Verlag, second edition, 2009.