Computational Fluid Dynamics

Prof. Dongwook Lee
dlee79@ucsc.edu

Figure 1. Some early pioneers of CFD in the era since WWII. Top level: Jay Boris, Vladimir Kolgan, Bram van Leer, Antony Jameson. Ground level: Richard Courant, Kurt Friedrichs, Hans Lewy, Robert MacCormack, Philip Roe, John von Neumann, Stanley Osher, Amir Harten, Peter Lax, Sergei Godunov. Courtesy of Bram van Leer.

Contents

Chapter 1

Fundamentals of CFD

1. What is CFD? Why do we study CFD?

Let's begin our first class with a couple of interesting scenarios.

Scenario 1: See Fig. 1. Consider you're a chief scientist in a big aerospace research lab. You're given a mission to develop a new aerospace plane that can reach at hypersonic speed (> Mach 5) within minutes after taking off. Its powerful supersonic combustion ramjets continue to propel the aircraft even faster to reach to a velocity near 26,000 ft/s (or 7.92 km/s, or Mach 25.4 in air at high altitudes, or a speed of NY to LA in 10 min), which is simply a low Earth orbital speed. This is the concept of transatmospheric vehicle the subject of study in several countries during the 1980s and 1990s. When designing such extreme hypersonic vehicles, it is very important to understand full three-dimensional flow filed over the vehicle with great accuracy and reliability. Unfortunately, ground test facilities – wind tunnels – do not exist in all the flight regimes around such hypersonic flight. We neither have no wind tunnels that can simultaneously simulate the higher Mach numbers and high flow field temperatures to be encountered by transatmospheric vehicles.

Scenario 2: See Fig. 2. Consider you're a theoretical astrophysicist who tries to understand core collapse supernova explosions. The theory tells us that very massive starts can undergo core collapse when the core fail to sustain against its own gravity due to unstable behavior of nuclear fusion. We simply cannot find any ground facilities that allow us to conduct any laboratory experiments in such highly extreme energetic astrophysical conditions. It is also true that in many astrophysical circumstances, both temporal and spatial scales are too huge to be operated in laboratory environments.

Scenario 3: See Fig. 3. Consider you a golf ball manufacturer. Your goal is to understand flow behaviors over a flying golf ball in order to make a better golf ball design (and become a millionaire!) Although you've already collected a wide range of the laboratory experimental data on a set of golf ball shapes (i.e., surface dimple design), you realize that it is very hard to analyze the data and understand them because the data are all nonlinearly coupled and can't

Figure 1. DARPA's Falcon HTV-2 unmanned aircraft can max out at a speed of about 16,700 miles per hour – Mach 22, NY to LA in 12 minutes.

be isolated easily. To keep your study in a better organized way, you wish to perform a set of parameter studies by controlling flow properties one by one so that you can also make reliable flow prediction for a new golf ball design.

As briefly hinted above, in practice there are various levels of difficulties encountered in real experimental setups. When performing the above mentioned research work, CFD therefore can be the major player that leads you to success because you obtain mathematical controls in numerical simulations. Let us take an example how numerical experiment via CFD can elucidate physical aspects of a real flow field. Consider the subsonic compressible flow over an airfoil. We are interested in answering the differences between laminar and turbulent flow over the airfoil for $\mathrm{Re} = 10^5$. For the computer program (assuming the computer algorithm is already well established, validated and verified!), this is a straightforward matter – it is just a problem of making one run with the turbulence model switched off (for the laminar setup), another run with the turbulence model switched on (for the turbulent flow), followed by a comparison study of the two simulation results. In this way one can mimic Mother Nature with simple knobs in the computer program – something you cannot achieve quite readily (if at all) in the wind tunnel. Without doubt, however, in order to achieve such success using CFD, you'd better to know what you do exactly when it comes to numerical modeling – the main goal of this course.

We are now ready to define what CFD is. CFD is a scientific tool, similar to experimental tools, used to gain greater physical insights into problems of interest. It is a study of the numerical solving of PDEs on a discretized system that, given the available computer resources, best approximates the real

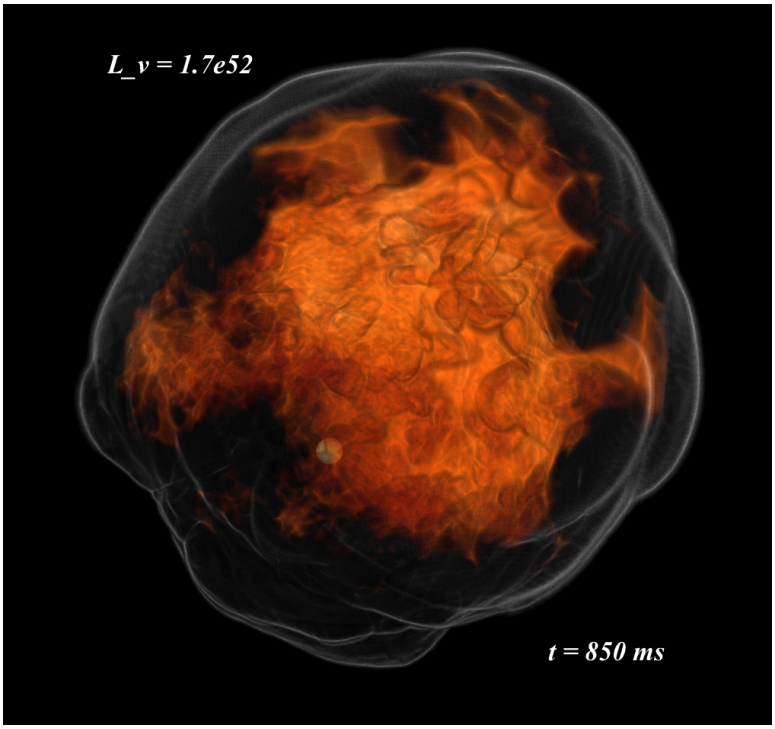


Figure 2. FLASH simulations of neutrino-driven core-collapse supernova explosions. Sean Couch (ApJ, 775, 35 (2013)).

geometry and fluid flow phenomena of interests. CFD constitutes a new "third approach" in studying and developing the whole discipline of fluid dynamics. A brief history on fluid dynamics says that the foundations for *experimental* fluid dynamics began in 17th century in England and France. In the 18th and 19th centuries in Europe, there was the gradual development of *theoretical* fluid dynamics. These two branches – experiment and theory – of fluid dynamics have been the mainstreams throughout most of the twentieth century. However, with the advent of the high speed computer with the development of solid numerical studies, solving physical models using computer simulations has revolutionized the way we study and practice fluid dynamics today – the approach of CFD. As sketched in Fig. 4, CFD plays a truly important role in modern physics as an equal partner with theory and experiment, in that it helps bringing deeper physical insights in theory as well as help better desiging experimental setups.

The real-world applications of CFD are to those problems that do *not* have known analytical solutions; rather, CFD is a scientific vehicle for solving flow problems that cannot be solved in any other way. In this reason – the fact that we use CFD to tackle to solve those *unknown* systems – we are strongly encouraged to learn thorough aspects in *all* three essential areas of study: (i) numerical theories, (ii) fluid dynamics, and (iii) computer programing skills.

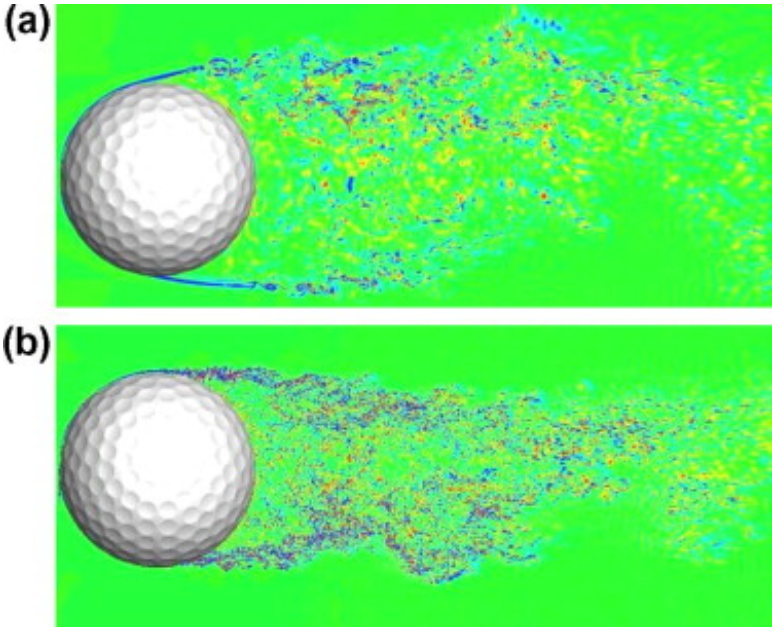


Figure 3. Contours of azimuthal velocity over a golf ball: (a) $\mathrm{Re} = 2.5 \times 10^4$; (b) $\mathrm{Re} = 1.1 \times 10^5$. C. E. Smith et al. (Int. J. Heat and Fluid Flow, 31, 262-273 (2010)).

2. The Governing Equations

In this chapter, we discuss fundamental principles in fluid dynamics and derive their governing equations, their physical meaning, and their mathematical forms particularly appropriate in CFD.

2.1. The fundamental equations of fluid dynamics

In modeling fluid motion, there are always following philosophy we need to consider. First is to choose the appropriate fundamental physical principles from the law of physics that are:

(a) Mass is conserved,
(b) $\mathbf{F} = \mathbf{ma}$ (Newton's second law), and
(c) energy is conserved.

We apply these physical principles to an appropriate flow model of our interest, and extract the needed mathematical equations which embody such physical principles. As we are interested in physical behaviors of a continuum fluid (or gas dynamics) in this course (rather than those of solid body, i.e., fluid mechanics rather than solid mechanics), we can construct one of the four models in modeling fluid motion:

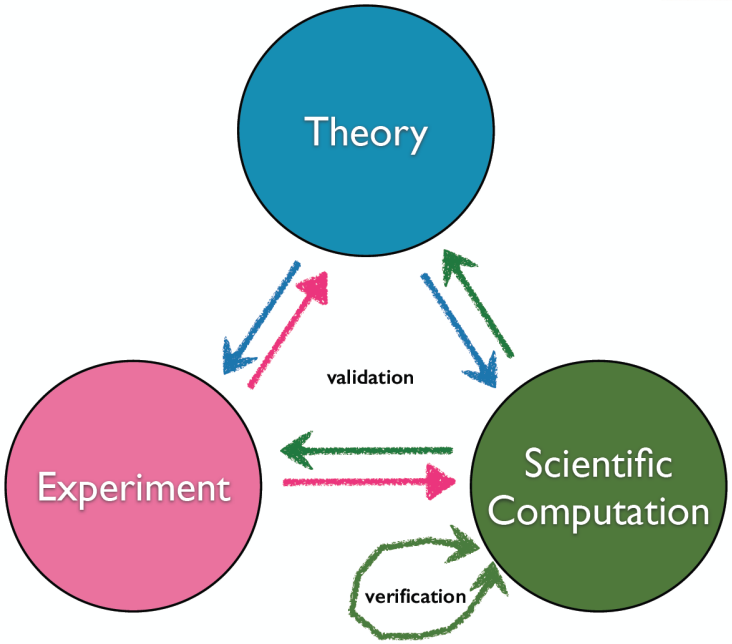


Figure 4. Three healthy cyclic relationship in fluid dynamics.

(F1) finite control volume approach fixed in space,
(F2) finite control volume approach moving with the fluid,
(F3) infinitesimal fluid element fixed in space, and finally,
(F4) infinitesimal fluid element fixed moving along a streamline.

The first two cases based on finite control volume (FCV) are illustrated in Fig. 5, whereas the last two cases of infinitesimal fluid element (IFE) are shown in Fig. 6. Let's now consider each of the four different approaches and derive the related mathematical relations.

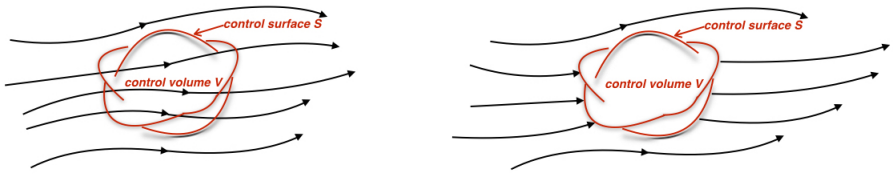


Figure 5. Finite control volume approach. Left: (F1) Finite control volume $\mathcal{V}$ fixed in space with the fluid moving through it. Right: (F2) Finite control volume moving $\mathcal{V}$ with the fluid with the same number of fluid particles kept in the same control volume $\mathcal{V}$.

2.1.1. General Remarks on FCV (F1 & F2):

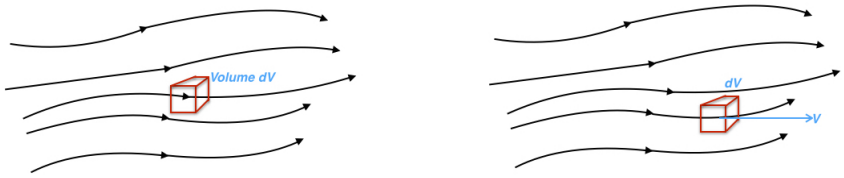


Figure 6. Infinitesimal fluid element approach. Left: (F3) Infinitesimal fluid element $d\mathcal{V}$ fixed in space with the fluid moving through it. Right: (F4) Infinitesimal fluid element $d\mathcal{V}$ moving along a streamline with the local velocity $\mathbf{V}$ equal to the local flow velocity at each point.

- We conceptually define 'FCV' a reasonably large closed region of the flow with a finite volume $\mathcal{V}$ and call its surface a 'control surface' S.
- FCV can be put in two different cases: (1) fixed in space with the fluid moving through it – this approach gives rise to the *conservative form* of the governing equations in *integral form*; (2) moving with the fluid such that the same fluid particles are always inside it – this results in the *nonconservative form* of the governing equations *integral form*.
- With the FCV approach, we limit our attention to just the fluid in the finite region of the volume itself (that is, we apply the law of physics to V) instead of looking at the whole flow field at once.

2.1.2. General Remarks on IFE (F3 & F4):

- In this approach we consider an infinitesimally small fluid element in the flow with a differential volume $d\mathcal{V}$.
- The fluid element is infinitesimal in the same sense as differential calculus and is large enough to contain a huge number of molecules (i.e., a continuous medium).
- As in FCV, two approaches are available wherein (3) IFE is fixed in space with the fluid moving through it – *conservative form* in *differential form* of the governing equations; and (4) moving along a streamline with a velocity vector $\mathbf{V}$ equal to the flow velocity at each point – *nonconservative form* of the *differential form* of the governing equations.

Note: We can possibly think of another approach that is based on the fundamental physics applied directly to the atoms and molecules – this is called the *kinetic theory* that solves the Boltzmann equations for individual particle using their distribution functions f_α. Notice that this approach has a microscopic view point in fluid motions, whereas FCV and IFE have a macroscopic view point.

2.2. Two important mathematical relations: D/Dt and $\nabla \cdot \mathbf{V}$

Before we start deriving the above mentioned mathematical relations, let's first take a moment to refresh our physical insights into two important mathematical

relations: (i) the substantial derivative D/Dt, and (ii) the divergence of velocity fields, $\nabla \cdot \mathbf{V}$.

(i) The substantial derivative D/Dt: Consider adopting the flow model described in F4, which is shown again in Fig. 7 in two different incidents in space and time in Cartesian space. Let's take a velocity vector $\mathbf{V} = u\mathbf{i} + v\mathbf{j} + w\mathbf{k}$, where each component is a function of both space and time,

$$u = u(x, y, z, t), \tag{1.1}$$
$$v = v(x, y, z, t), \tag{1.2}$$
$$w = w(x, y, z, t). \tag{1.3}$$

We denote the scalar density field by

$$\rho = \rho(x, y, z, t). \tag{1.4}$$

The density of the *same* fluid at the two different locations of space and time

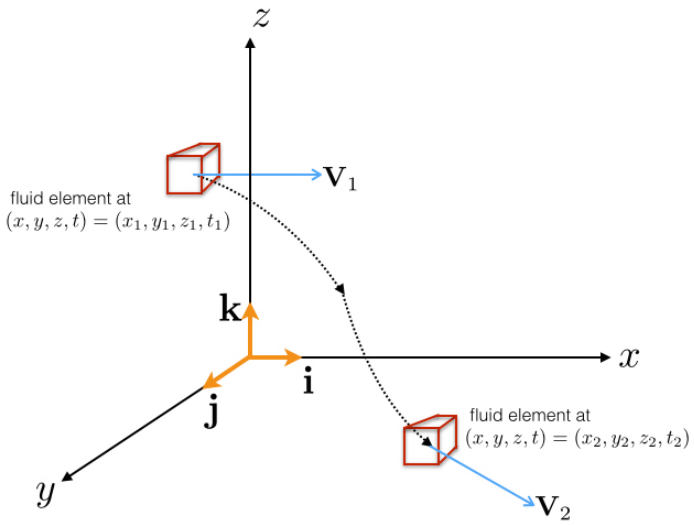


Figure 7. Illustration for the substantial derivative for a fluid element moving in the flow

can be written as $\rho_1 = \rho(x_1, y_1, z_1, t_1)$ and $\rho_2 = \rho(x_2, y_2, z_2, t_2)$, where we can further expand the density function about point 1 as follows:

$$\rho_2 = \rho_1 + \left(\frac{\partial \rho}{\partial x}\right)_1 (x_2 - x_1) + \left(\frac{\partial \rho}{\partial y}\right)_1 (y_2 - y_1) + \left(\frac{\partial \rho}{\partial z}\right)_1 (z_2 - z_1) + \left(\frac{\partial \rho}{\partial t}\right)_1 (t_2 - t_1) + H.O.T \tag{1.5}$$

Dividing by $t_2 - t_1$ and ignoring high-order terms (H.O.T), we get

$$\frac{\rho_2 - \rho_1}{t_2 - t_1} = \left(\frac{\partial \rho}{\partial x}\right)_1 \frac{x_2 - x_1}{t_2 - t_1} + \left(\frac{\partial \rho}{\partial y}\right)_1 \frac{y_2 - y_1}{t_2 - t_1} + \left(\frac{\partial \rho}{\partial z}\right)_1 \frac{z_2 - z_1}{t_2 - t_1} + \left(\frac{\partial \rho}{\partial t}\right)_1 \tag{1.6}$$

Take a look at the LHS of Eq. 1.6 and we notice that this is physically the 'average' time rate of change in density of the fluid element as it moves from point 1 to point 2. in the limit of $t_2 \to t_1$, we get

$$\lim_{t_2 \to t_1} \frac{\rho_2 - \rho_1}{t_2 - t_1} \equiv \frac{D\rho}{Dt} \tag{1.7}$$

By definition, the symbol is called the substantial derivative D/Dt and it has its physical meaning that measures the time rate of change of a given quantity (density in our current example) of the given fluid element as it moves from one location to another in both space and time.

Note: Notice that there is a clear difference between D/Dt and $\partial/\partial t$ in that the latter is called the *local derivative* which represents the time rate of change at a 'fixed' point – our eyes are locked on the stationary point 1; whereas for the first, our eyes are locked on the fluid element as it moves watching its density change as it passes through point 1.

Now, taking the limit of Eq. 1.6 as $t_2 \to t_1$, we can further cast the relation into

$$\frac{D\rho}{Dt} = u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} + \frac{\partial \rho}{\partial t} \tag{1.8}$$

Finally, we can obtain an expression for the substantial derivative in Cartesian coordinate system:

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} + w\frac{\partial}{\partial z} = \frac{\partial}{\partial t} + \mathbf{V} \cdot \nabla, \tag{1.9}$$

where we have introduced

$$\nabla \equiv \mathbf{i}\frac{\partial}{\partial x} + \mathbf{j}\frac{\partial}{\partial y} + \mathbf{k}\frac{\partial}{\partial z}. \tag{1.10}$$

Quick summary:

- D/Dt is called the *substantial derivative* (or, also called *material derivative*),
- $\partial/\partial t$ is called the *local derivative*, and
- $\mathbf{V} \cdot \nabla$ is called the *convective derivative*.

Note: Recall that the substantial derivative is nothing but a total derivative with respect to time, d/dt. In other words, from differential calculus, we easily see that

$$d\rho = \frac{\partial \rho}{\partial x}dx + \frac{\partial \rho}{\partial y}dy + \frac{\partial \rho}{\partial z}dz + \frac{\partial \rho}{\partial t}dt, \tag{1.11}$$

which yields

$$\frac{d\rho}{dt} = u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z} + \frac{\partial \rho}{\partial t} \tag{1.12}$$

Example: You are entering an ice cave with a friend of yours. You will experience a temperature decrease as you walk deeper in to the cave – this is analogous to the convective derivative. As you keep walking in to the cave, your friend throws a snowball at you and you feel an additional instantaneous temperature drop when the snowball hits you – this effect is analogous to the local derivative. Notice that the substantial derivative is the sum of the two effects.

(ii) The divergence of the velocity fields $\nabla \cdot \mathbf{V}$: Consider a finite control volume (FCV) moving from one place to another depicted as in Fig. 8. In this example, the FCV is consist of the same fluid particles when moving, therefore keeping is mass fixed in time. However, its volume $\mathcal{V}$ and its control surface S can vary with time as it moves to a different location of the flow where different density occupies. That is, the control volume keeps changing its volume and shape depending on the characteristic of the flow.

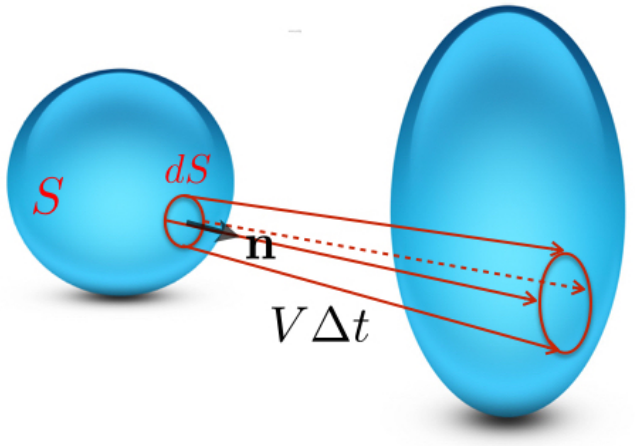


Figure 8. Moving control volume for the physical interpretation the divergence of the velocity fields

Let us now focus on an infinitesimal element of surface dS moving at the local velocity $\mathbf{V}$ along the normal direction $\mathbf{n}$ which is perpendicular to dS. The change in the volume $\Delta\mathcal{V}$ of the control volume due to the movement of dS over Δt is available by inspecting the volume of the long, thin cylinder with the base area of dS and the height $\mathbf{V}\Delta t \cdot \mathbf{n}$. That is,

$$\Delta\mathcal{V} = \mathbf{V}\Delta t \cdot \mathbf{n}dS = \mathbf{V}\Delta t \cdot \mathbf{dS}, \tag{1.13}$$

where $\mathbf{n}dS = \mathbf{dS}$. In the limit of $dS \to 0$, the total change in volume of the whole control volume is

$$\int\int_S \mathbf{V}\Delta t \cdot \mathbf{dS}. \tag{1.14}$$

After dividing Eq. 1.14 by Δt and subsequently apply the divergence theorem, we obtain its physical meaning of 'the time rate of change of the control volume',

denoted by $\frac{D\mathcal{V}}{Dt}$ (note here that we used the substantial derivative notation of $\mathcal{V}$ as we wish to define the time rate of change of the control volume *as the volume moves along with the flow*):

$$\frac{D\mathcal{V}}{Dt} = \frac{1}{\Delta t}\int\int_S \mathbf{V}\Delta t \cdot \mathbf{dS} = \int\int_S \mathbf{V}\cdot\mathbf{dS} = \int\int\int_{\mathcal{V}} \nabla\cdot\mathbf{V}d\mathcal{V}. \tag{1.15}$$

By keeping continuously shrink $\mathcal{V}$ to $\delta\mathcal{V}$ in such a way that $\delta\mathcal{V}$ is so small enough to treat $\nabla\cdot\mathbf{V}$ as constant in $\delta\mathcal{V}$. Then in the limit of $\delta\mathcal{V}\to 0$, we can rewrite Eq. 1.15 as

$$\frac{D(\delta\mathcal{V})}{Dt} = \int\int\int_{\delta\mathcal{V}} \nabla\cdot\mathbf{V}d\mathcal{V} = \nabla\cdot\mathbf{V}\delta\mathcal{V}, \tag{1.16}$$

or

$$\nabla\cdot\mathbf{V} = \frac{1}{\delta\mathcal{V}}\frac{D(\delta\mathcal{V})}{Dt} \tag{1.17}$$

Quick summary:

- $\nabla\cdot\mathbf{V}$ physically means the time rate of change of the volume of a moving fluid element per unit volume.

2.3. The Continuity Equation

We are now ready to apply the philosophy discussed in Sec. 2.1. to all four of the flow models illustrated in Figs. 5 and 6. Let's begin with the first principle:

(a) Mass is conserved.

We are going to derive the continuity equation in four different ways and see they are all related mathematically.

(F1) FCV fixed in space:
Let us examine the principle of the mass conservation by considering a small control volume $\mathcal{V}$ surrounded by its control surface S as depicted in the left panel of Fig. 5. Then the mass conservation law can be stated as:

`The net mass flow 'out' of` $\mathcal{V}$ `through surface` S = `The time rate of 'decrease' of mass inside` $\mathcal{V}$

In order to obtain a mathematical expression for LHS, we write the mass flow of a moving fluid with fluid velocity $\mathbf{V}$ across any fixed surface. The elemental mass flow across the area dS normal to $\mathbf{n}$ becomes

$$\rho\mathbf{V}\cdot\mathbf{n}dS = \rho\mathbf{V}\cdot\mathbf{dS} \tag{1.18}$$

Recall that by convention, the direction of the flow is 'out' of $\mathcal{V}$ because $\mathbf{dS}$ points in a direction 'out' of $\mathcal{V}$, hence the mass inside $\mathcal{V}$ 'decreases' in the above statement. By taking the surface integral of Eq. 1.18, we obtain the net mass flow out of the entire control volume $\mathcal{V}$ – the expression for LHS:

$$\int\int_S \rho\mathbf{V}\cdot\mathbf{dS} \tag{1.19}$$

The expression for RHS is the time rate of 'decrease' of the total mass $\int\int\int_{\mathcal{V}} \rho d\mathcal{V}$ inside $\mathcal{V}$, that is,

$$-\frac{\partial}{\partial t}\int\int\int_{\mathcal{V}} \rho d\mathcal{V} \tag{1.20}$$

Equating the two, we finally get a mathematical relation for the mass conservation:

$$\frac{\partial}{\partial t}\int\int\int_{\mathcal{V}} \rho d\mathcal{V} + \int\int_{S} \rho \mathbf{V}\cdot\mathbf{dS} = 0 \tag{1.21}$$

Note: We emphasize that Eq. 1.21 is an *integral form of the continuity equation*. The 'finite' aspect of the control volume is why the equation is obtained directly in integral form. The fact that the control volume was 'fixed in space' resulted in the specific integral form given by Eq. 1.21, which is called the *conservation form*.

(F2) FCV moving with the fluid: As seen earlier, we can write another mathematical expression for the mass conservation law using the substantial derivative which perfectly describes behavior of the time rate of change of any property of a fluid element moving with the flow. That is to say, the mass conservation law is simply put into a form

$$\frac{D}{Dt}\int\int\int_{\mathcal{V}} \rho d\mathcal{V} = 0 \tag{1.22}$$

Note: We remark that Eq. 1.22 is also an *integral form of the continuity equation* which is different from the previous result – this is now called the *nonconservation form*. Comparing with the previous conservation form, we can see that the nonconservative form is a result of considering the control volume *moving* with the fluid.

(F3) IFE fixed in space: For convenience we adopt an infinitesimal fluid element fixed in space in a Cartesian coordinate system shown in Fig. 9. What we want to calculate is the net mass flow through all surrounding six faces with the elemental areas of $dxdy$, $dydz$ and $dxdz$. As illustrated in Fig. 9, we consider each individual net flow in each coordinate direction. They are
(a) the net outflow in x-direction:

$$(\rho u + \frac{\partial \rho u}{\partial x}dx)dydz - (\rho u)dydz = \frac{\partial \rho u}{\partial x}dxdydz, \tag{1.23}$$

(b) the net outflow in y-direction:

$$(\rho v + \frac{\partial \rho v}{\partial y}dy)dxdz - (\rho v)dydz = \frac{\partial \rho v}{\partial y}dxdydz, \tag{1.24}$$

(c) the net outflow in z-direction:

$$(\rho w + \frac{\partial \rho w}{\partial z}dz)dxdy - (\rho w)dxdy = \frac{\partial \rho w}{\partial z}dxdydz. \tag{1.25}$$

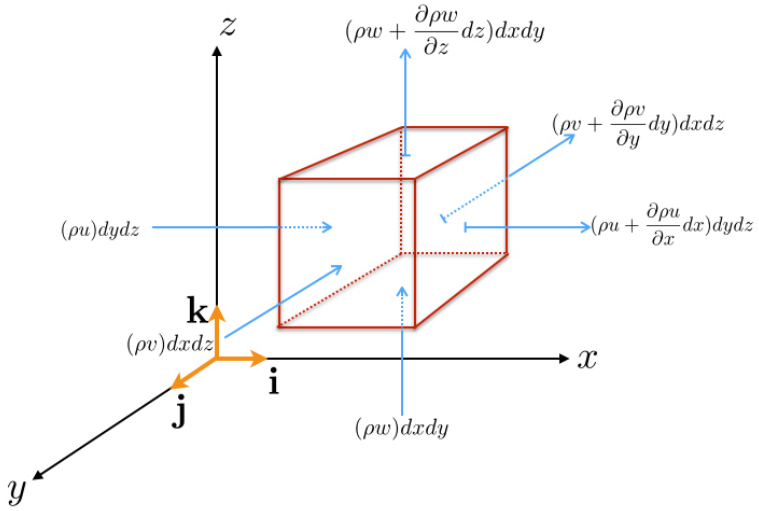


Figure 9. Model of the infinitesimal fluid element fixed in space and mass fluxes through various faces of the element

Hence, the net mass flow out of the element in all directions is given by summing all of the above relations:

$$\Big(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z}\Big) dxdydz, \tag{1.26}$$

which should be equal to the time rate of decrease of the total mass $\rho dxdydz$ in the infinitesimal element of volume $dxdydz$:

$$-\frac{\partial \rho}{\partial t} dxdydz \tag{1.27}$$

Equating the two we get yet another form describing the mass conservation

$$\frac{\partial \rho}{\partial t} + \Big(\frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} + \frac{\partial \rho w}{\partial z}\Big) = \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \tag{1.28}$$

Note: We call Eq. 1.28 the *differential form of the continuity equation in conservation form*. The 'infinitesimal' aspect of the small element lead to the differential form of the equation, and, as before, the fact that the fluid element was 'fixed in space' resulted in the *conservation form*.

(F4) IFE moving with the fluid: We remind ourselves that although the mass of an IFE is conserved when it moves with the fluid, its elemental volume $\delta\mathcal{V}$ varies. Since the mass in the IFE is invariant, invoking the physical meaning of the substantial derivative and using the chain rule, we have

$$0 = \frac{D\rho\delta\mathcal{V}}{Dt} = \delta\mathcal{V}\frac{D\rho}{Dt} + \rho\frac{D\delta\mathcal{V}}{Dt} \tag{1.29}$$

Combining the definition of the divergence of the velocity fields in Eq. 1.17, this can be rewritten as

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{V} = 0 \tag{1.30}$$

Note: We call Eq. 1.30 the *differential form of the continuity equation in nonconservation form*. The 'infinitesimal' aspect of the small element lead to the differential form of the equation, while the fact that the fluid element was 'moving with the fluid' resulted in the *nonconservation form* as in (F2).

Problem 1. Often times, the condition for incompressible flows is given by $\nabla \cdot \mathbf{V} = 0$. Why?

Problem 2. Derive the integral form of the momentum equation in conservation form

$$\frac{\partial}{\partial t}\int\int\int_{\mathcal{V}} \rho\mathbf{V}d\mathcal{V} + \int\int_S (\rho\mathbf{V}\cdot\mathbf{dS})\mathbf{V} = \int\int\int_{\mathcal{V}} \rho\mathbf{f}d\mathcal{V} - \int\int_S p\mathbf{dS} \tag{1.31}$$

using the Newton's second law applied to a fluid flow. Ignore any viscous effect (Hint: $\mathbf{F} = \frac{d}{dt}(m\mathbf{V})$).

Problem 3. Derive the integral form of the energy equation in conservation form using the energy conservation law for adiabatic inviscid flows:

$$\frac{\partial}{\partial t}\int\int\int_{\mathcal{V}} \rho(e+\frac{V^2}{2})d\mathcal{V} + \int\int_S \rho(e+\frac{V^2}{2})\mathbf{V}\cdot\mathbf{dS} = \int\int\int_{\mathcal{V}} \rho\mathbf{f}\cdot\mathbf{V}d\mathcal{V} - \int\int_S p\mathbf{V}\cdot\mathbf{dS} \tag{1.32}$$

Problem 4. Show that all four approaches discussed in (F1)-(F4) for the continuity equation are in fact all the same. That is, one of them can be obtained from any of the others. (Hint: You can show that there are equivalent relationships in circle: (F1) $\Rightarrow$ (F2) $\Rightarrow$ (F4) $\Rightarrow$ (F3) $\Rightarrow$ (F1))

Chapter 2

Scalar Conservation Laws - Theories

In many practical applications of CFD, one mostly tackles physical phenomena described by 'systems' of (nonlinear) equations such as the Euler or Navier-Stokes equations. Solving such systems is more complicated than solving a scalar equation (linear or nonlinear) in both mathematical and computational aspects.

However, we often gain rich insights in our understandings of the more complicated systems from studying the simpler systems first. In this chapter, we seek for a good understanding of the linear and nonlinear scalar advection equations, whereby it will enlighten us in achieving our bigger goals in studying the systems of (nonlinear) conservation laws later.

1. Linear scalar equations

We consider two types of linear scalar advection equations, one with a constant velocity a, and the other with a variable velocity $a(t)$, where $x = x(t)$. Let's first take a look at the 1D linear scalar advection equation for $t \geq 0$ written as

$$u_t + au_x = 0 \tag{2.1}$$

with a constant advection velocity a, and together with initial conditions on $\mathbb{R}$,

$$u(x, 0) = u_0(x). \tag{2.2}$$

As shown in the previous chapter, we know the solution is given by

$$u(x, t) = u_0(x - at) \tag{2.3}$$

for $t \geq 0$. Recall that $x - at = x_0$ is called the characteristic line with a given constant x_0 and with the propagation velocity a. Depending on the sign of a, the initial data $u_0(x)$ is advected (or transported) – hence the name 'advection equation' – to the right (if $a > 0$) or left (if $a < 0$). Note that there are infinitely many characteristic lines emanating from the initial condition in the x-t plane as there are infinite choices of $x_0 \in \mathbb{R}$. See Fig. 1.

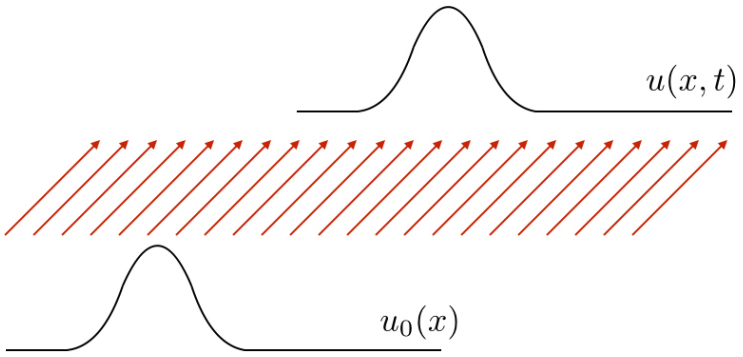


Figure 1. Characteristic curves and the advection of the solution. All information is simply advected to the later time solution $u(x,t)$ for $t>0$ (top) along the characteristic curves in the x-t plane *without* any shape changes from the initial condition $u(x,t=0)=u_0(x)$ (bottom).

In general, the characteristics are curves (or simply 'the characteristics') in the x-t plane satisfying the ODEs

$$x'(t)=a \text{ and } x(0)=x_0. \tag{2.4}$$

One very important property on the characteristics is that the solution $u(x,t)$ of the constant velocity a remains as constant along the characteristics. To see this,

$$\frac{d}{dt}u(x(t),t)=\frac{\partial}{\partial t}u(x(t),t)+\frac{\partial}{\partial x}u(x(t),t)x'(t)=u_t+au_x=0, \tag{2.5}$$

confirming the claim.

In the more general case of the scalar equation with the variable velocity $a(x(t))$, we consider

$$u_t+\Big(a(x(t))u\Big)_x=0. \tag{2.6}$$

In this case, the characteristics are no longer straight lines satisfying

$$x'(t)=a(x(t)) \text{ and } x(0)=x_0, \tag{2.7}$$

and the solution $u(x,t)$ is no longer constant along the characteristics. This can be easily verified if we rewrite Eq. 2.6 as

$$u_t+a(x(t))u_x=-a'(x(t))u, \tag{2.8}$$

therefore we obtain

$$\frac{d}{dt}u(x(t),t)=-a'(x(t))u\neq 0. \tag{2.9}$$

In both cases of the constant and variable velocities, the solution can be easily determined by solving sets of ODEs.

Remark: In words, the characteristic curves track the motion of material particles.

Remark: We can see that if $u_0(x) \in C^k(\mathbb{R})$ then $u(x,t) \in C^k(\mathbb{R}) \times (0,\infty)$.

Remark: So far, we have assumed differentiability of $u(x,t)$ in manipulating the above relations. Note that this assumption makes it possible to seeks for a classical solution $u(x,t)$ of the differential equations.

1.1. Domain of dependence & Range of influence

We now make an important observation in solutions to the linear advection equations:

The solution $u(x,t)$ at any point $(\bar{x},\bar{t})$ depends only on the initial data u_0 only at a *single* point, namely $\bar{x}_0$ such that $(\bar{x},\bar{t})$ lies on the characteristic through $\bar{x}_0$.

This means that the solution $u(\bar{x},\bar{t})$ will remain unchanged no matter how we change the initial data at any points other then $\bar{x}_0$. We now define two related regions, the first is called the domain of dependence, and the second is called the range of influence.

Definition: The set $\bar{\mathcal{D}}(\bar{x},\bar{t}) = \{\bar{x} - \lambda_m \bar{t} : m = 1,2,...,p\}$ is called the domain of dependence of the point $(\bar{x},\bar{t})$, where p is the total number of characteristic velocities (or the number of equations of hyperbolic PDE systems). See Fig. 2 for an illustration.

Remark: For convenience, let us assume $\lambda_1 \leq ... \leq \lambda_m \leq ... \leq \lambda_p$. Note that $p = 1$ for *scalar* hyperbolic equations, whereas $p > 1$ for *systems* of hyperbolic equations. For instance, $p = 3$ for the systems of 1D Euler equations (1 continuity equations, 1 momentum equation, and 1 energy equation).

Note: What are the values of p for the systems of 2D Euler and 3D Euler equations?

Definition: The region $\mathcal{R} = \{x : \lambda_1 t \leq x - x_0 \leq \lambda_p t\}$ is called the range of influence of the point x_0. See Fig. 3 for an illustration.

Note: One can always find a bounded set $\mathcal{D} = \{x : |x - \bar{x}| \leq \lambda_p \bar{t}\}$ such that $\bar{\mathcal{D}}(\bar{x},\bar{t}) \subset \mathcal{D}$. The existence of $\bar{\mathcal{D}}$ and $\mathcal{R}$ are the consequence of the fact that hyperbolic equations have *finite propagation speed*; information can travel with speed at most

$$\max_m \{|\lambda_m| : m = 1,...,p\}$$

.

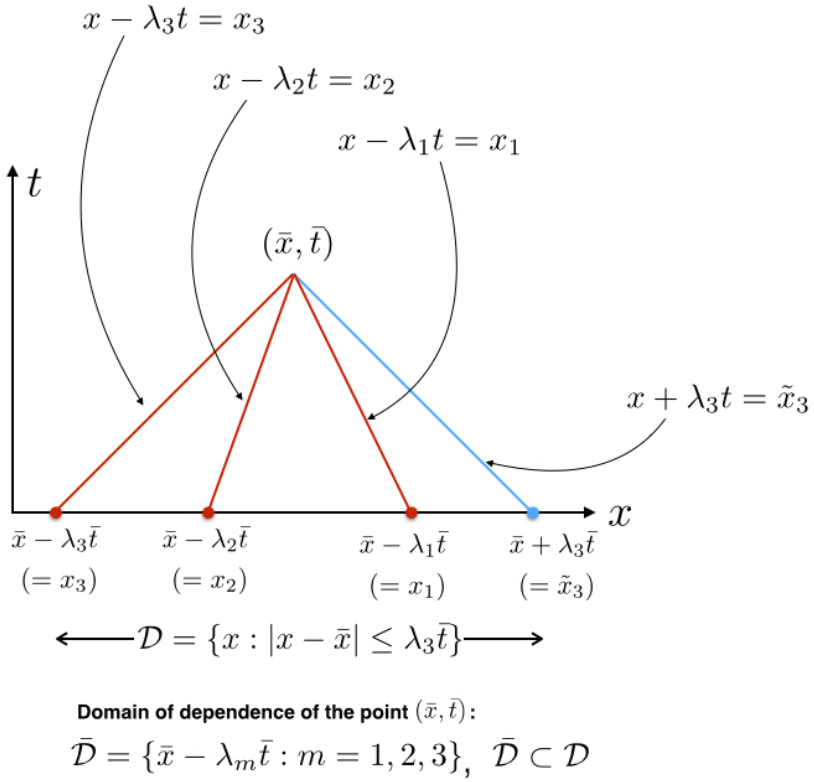


Figure 2. The domain of dependence of the point $(\bar{x}, \bar{t})$ for a typical hyperbolic system of three equations with $\lambda_1 < 0 < \lambda_2 < \lambda_3$. Note that one can always find a *bounded* domain $\mathcal{D}$ such that $\bar{\mathcal{D}} \subset \mathcal{D}$ because of the fact that the propagation velocities (or characteristic velocities) of hyperbolic PDEs are always finite.

1.2. Non-smooth data

Consider for a moment what happens if $u_0(x)$ has a singularity at some point x_0 (i.e., a discontinuity in u_0 or some derivatives). In this case, the resulting $u(x,t)$ will have a singularity of the same order along the characteristic curve though x_0. This is a fundamental property of *linear* hyperbolic equations in which singularities are simply advected only along characteristics (Also see Fig. 2). This is because the solution, $u(x,t) = u_0(x - at)$, along a characteristic curve $x - at = x_0$, only depends on the one and only value $u_0(x_0)$, thus allowing a non-smooth"solution" to the PDE even if $u_0(x)$ is not smooth.

Such non-smooth solution, although it is no longer a classical solution of the differential equation everywhere, *does* satisfy the integral form of the conservation law, which continues to make sense for non-smooth u as long as u is an integrable function.

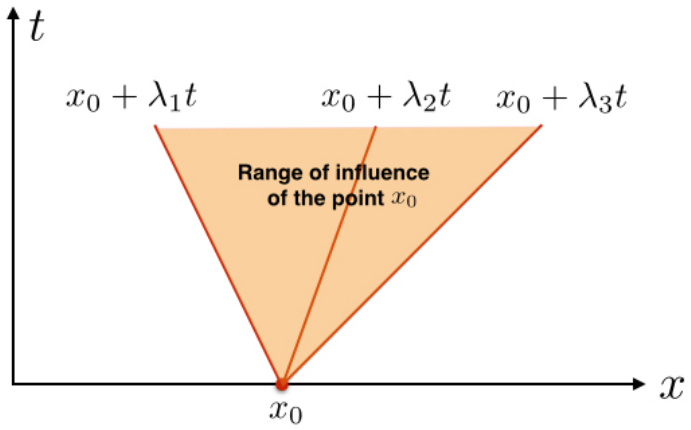


Figure 3. The range of influence $\mathcal{R} = \{x : \lambda_1 t \leq x - x_0 \leq \lambda_3 t\}$ of the point x_0 of the same problem in Fig. 2. Notice that the conic region $\mathcal{R}$ is a symmetric image of $\mathcal{D}$ with respect to $(\bar{x}, \bar{t})$, shifted to $t = 0$ axis.

Therefore, it may sound like a perfect idea to accept this concept – i.e., integrating along characteristics regardless of the regularity of $u_0(x)$ – in order to achieve a generalized solution $u(x,t)$. Unfortunately, we can no longer simply integrate along characteristics when solving the *nonlinear* equations (yes, the linear case is relatively too easy!) because the nonlinear characteristic curves often converge (collide) each other to form a shock, losing their characteristic information for good. The nonlinear equations also can develop singularities even from a smooth initial data $u_0(x)$.

One working idea that can be generalized to both linear and nonlinear equations, is to leave the initial data alone but modify the PDE by adding a small diffusive term ϵu_{xx} and take the limit of the diffusive term as $\epsilon \to 0$. The solution obtained this way is called the *vanishing viscosity* solution. Mathematically, one writes an advection-diffusion equation

$$u_t + a u_x = \epsilon u_{xx} \tag{2.10}$$

as an approximation to the advection equation for very small $\epsilon > 0$. Notice that we can always find the solution $u^\epsilon \in C^\infty(\mathbb{R}) \times \mathbb{R}^+$ to Eq. 2.10 even if $u_0(x)$ is not smooth, because Eq. 2.10 is a parabolic equation (why? See **Problem 1**). We can therefore obtain a generalized solution $u(x,t)$ by

$$\lim_{\epsilon \to 0} u^\epsilon(x,t) = u(x,t). \tag{2.11}$$

Problem 1 Use a change of variables to follow the characteristics (i.e., $\xi = x + at$ and $\tau = t$) and set

$$v^\epsilon(x,t) = u^\epsilon(x + at, t) = u^\epsilon(\xi, \tau) \tag{2.12}$$

(a) Assuming u^ϵ is a solution to Eq. 2.10 (i.e., $u^\epsilon_\tau + a u^\epsilon_\xi = \epsilon u^\epsilon_{\xi\xi}$), first show that v^ϵ satisfies the heat equation

$$v^\epsilon_t(x,t) = \epsilon v^\epsilon_{xx}(x,t). \tag{2.13}$$

Note that we have converted the advection-diffusion equation Eq. 2.10 to the pure diffusion equation Eq. 2.13. Now, using the well-known solution to the diffusion equation to solve for $v^\epsilon(x,t)$ (see Hint 1),
(b) Show that we have (this should be very trivial)

$$u^\epsilon(x,t) = v^\epsilon(x - at, t). \tag{2.14}$$

(c) And moreover, show that (use Hint 2)

$$\lim_{\epsilon\to 0} u^\epsilon(x,t) = \lim_{\epsilon\to 0} v^\epsilon(\xi - a\tau, \tau) = u_0(\xi - a\tau) \tag{2.15}$$

Hint 1: For the diffusion equation Eq. 2.13, we can always find the classical solution of the PDE using Green's functions:

$$v^\epsilon(x,t) = \frac{1}{\sqrt{4\pi\epsilon t}} \int_R e^{-\frac{(x-y)^2}{4\epsilon t}} v^\epsilon(y,0) dy, \tag{2.16}$$

where $v^\epsilon(y,0) = u^\epsilon(y,0) = u^\epsilon_0(y)$.
Hint 2: Let $g(x)$ be a bounded function and is continuous at $x = 0$. Let

$$\gamma_r(x) = \sqrt{\frac{r}{\pi}} e^{-rx^2}. \tag{2.17}$$

Then

$$\lim_{r\to\infty} \int_{\mathbb{R}} \gamma_r(x - y) g(x) dx = g(y). \tag{2.18}$$

Note that $\gamma_r(x)$ is Gaussian and has the following properties:
(a) $\gamma_r(x) \geq 0$,
(b) $\lim_{r\to\infty} \gamma_r(x) = 0$, if $x \neq 0$; $\lim_{r\to\infty} \gamma_r(x) = 0$, if $x = 0$,
(c) $\int_{\mathbb{R}} \gamma_r(x) dx = \sqrt{\frac{r}{\pi}} \int_{\mathbb{R}} e^{-rx^2} dx = \frac{1}{\sqrt{\pi}} \int_{\mathbb{R}} e^{-y^2} dy = 1$.

2. Nonlinear scalar equations

We now move on to consider the nonlinear scalar equation

$$u_t + \Big(f(u)\Big)_x = 0 \tag{2.19}$$

where $f(u)$ is a nonlinear function of u and is called the flux function. There are two types of flux functions that give rise to different solution behaviors, especially involving solution structures containing shocks and/or rarefaction waves:

1. $f(u)$ is a convex function – i.e., $f''(u) > 0$, $\forall u$ (or, equally well, f is concave with $f''(u) < 0$, $\forall u$): e.g., the Burger's equation, the Euler equations, the Navier-Stokes equations.

2. $f(u)$ is a non-convex function: e.g., the Buckley-Leverett equation, magnetohydrodynamics (MHD) equations.

Remark: Later, we will see that the "convexity" assumption in the nonlinear scalar equation corresponds to a "genuinely nonlinearity" assumption for systems of equations.

Definition: If we rewrite Eq. 2.19 in nonconservation form, we get $u_t + f'(u)f_x(u) = 0$. The derivative of the flux function

$$\lambda(u) = f'(u) = \frac{df}{du} \tag{2.20}$$

is called the *characteristic speed*.

Remark:

1. In the system case,

$$0 = \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{U}_t + \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x}, \tag{2.21}$$

   the characteristic speed corresponds to the eigenvalues of the Jacobian matrix $\frac{\partial \mathbf{F}}{\partial \mathbf{U}}$.

2. For the linear scalar advection case, we already saw that $\frac{df}{du} = a$.

By far the most famous and popular example in the nonlinear scalar equations is Burgers' equation, in which the flux function is given as

$$f(u) = \frac{u^2}{2}, \tag{2.22}$$

hence resulting in the equation in the nonconservation form as

$$u_t + uu_x = 0. \tag{2.23}$$

We now take a look at the its mathematical properties from two different perspectives: (i) for small t, and (ii) for large t.

2.1. Burgers' equation for small t_s

Let's first assume that the initial data $u_0(x)$ is smooth and no singularity is observed for $0 < t \leq t_s$. In this case, we can conveniently follow characteristics

$$x'(t) = u(x(t), t) \tag{2.24}$$

along which the solution $u(x, t)$ is constant, since

$$\frac{d}{dt}u(x(t), t) = \frac{\partial}{\partial t}u(x(t), t) + \frac{\partial}{\partial x}u(x(t), t)x'(t) = u_t + uu_x = 0. \tag{2.25}$$

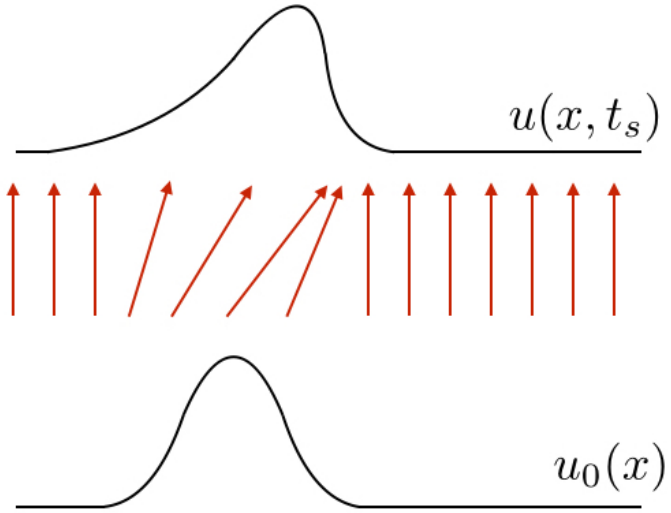


Figure 4. Characteristics and solution for Burgers' equation for small $t = t_s$.

This also tells us that the slope $x'(t)$ is constant, and so the characteristics are straight lines, determined by the initial data. See Fig. 4.

Therefore, if the initial data $u_0 = u(\xi, 0)$ is smooth, and if t_s is chosen small enough so that the characteristics do not cross each others, we can solve the equation

$$x = \xi + u(\xi, 0)t_s \tag{2.26}$$

for ξ, and thus we obtain a well-defined solution

$$u(x, t_s) = u(x - u(\xi, 0)t_s, 0). \tag{2.27}$$

2.2. Burgers' equation for large t_b: Shock formation

For large $t = t_b$ at or after which the characteristics cross, Eq. 2.26 may not have a unique solution. This indeed will occur if $u_0'(\xi) < 0$ at any point ξ – that is, if $u_0(\xi)$ is a monotone decreasing function of ξ then the characteristics $x(t) = \xi(t) + u_0(\xi(t))t$ eventually cross at some finite time $t = t_b$ at which the wave will break and develop into a shock. When this first happens at $t = t_b$, the function $u(x, t)$ has an infinite slope, beyond which there is no classical solution of the PDE, and the (weak) solution becomes discontinuous. See Fig. 5.

Problem 2 Given a smooth initial data $u_0(\xi)$ for Burgers' equation with its slope $u_0'(\xi) < 0$ at some point ξ_0. Show that the wave break time t_b is written as

$$t_b = \frac{-1}{u_0'(\xi_0)}. \tag{2.28}$$

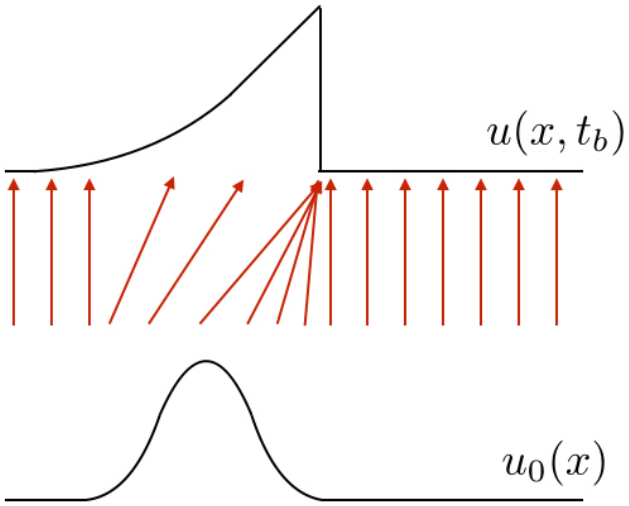


Figure 5. Characteristics and solution for Burgers' equation for large $t = t_b$. The characteristics cross and a shock is formed as a result.

Recall that in the case of the linear scalar advection, in which the characteristic speed is constant, $df/du = a$, the solution is simply a translated form of the initial data with speed a without any distortion (see Fig. 1). In the nonlinear case the characteristic speed is a function of the solution $u(x, t)$ itself – e.g., $df/du = u$ for Burgers' equation, therefore, distortions are inevitably produced. This is a distinctive feature of nonlinear problem.

To see the wave distortion phenomenon – also referred to as 'the wave steepening' – we refer to the initial condition $u_0(x)$ depicted as in Figs. 4 & 5. First, note that the flux function for Burgers' equation is convex (i.e., $f''(u) = 1 > 0$), and therefore, its characteristic speed (i.e., $f'(u) = u$) is an increasing function of u – the characteristic speed of Burgers' equation is u itself. The behavior of the characteristic speed therefore depends on the behavior of u. Specifically, the initial characteristics $x_m(t)$ emanating from the initial points $\bar{x}_m$, $m = 1, ..., p$ have the form (see also Fig. 3)

$$x_m(t) = \bar{x}_m + u_0(\bar{x}_m)t. \tag{2.29}$$

We see that depending on how $u_0(x)$ increases or decreases as a function of x, the initial characteristic speeds vary, and the characteristic curves can cross. We can think of two intervals I_e and I_c on the x-axis where distortions are more evident. See Fig. 6 for an illustration. If we let $\bar{x}_0$ to be a point where $u_0'(\bar{x}_0) = 0$ (i.e., $\bar{x}_0$ is a local maximum point of u_0), and take

$$I_e = [\bar{x}_+, \bar{x}_0], I_c = [\bar{x}_0, \bar{x}_-], \tag{2.30}$$

where $\bar{x}_+$ and $\bar{x}_-$ are the points where u_0 starts to increase and stops to decrease as x, respectively, as shown in Fig. 6. We say that I_e is an expansive region

where the characteristic speed keeps increasing as x increases. On the contrary, I_c is a compressive region where the characteristic speed decreases with x. It is easy to see that the characteristics from I_e and I_c will eventually cross each others, generating a sharp discontinuous profile of $u(x,t)$ for $t > t_b$, although the initial data $u_0(x)$ was smooth to begin with.

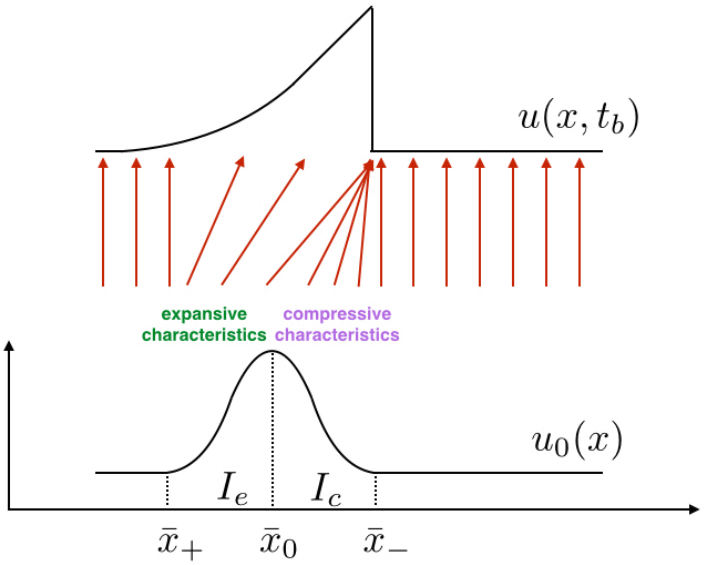


Figure 6. Characteristics crossing for Burgers' equation for large $t > t_b$.

For times $t > t_b$ some of the characteristics have crossed. When this happens, there are points x where there are three characteristics leading back to $t = 0$. The solution u at such a time is a triple-valued function as seen in Fig. 7. Although there exist some cases that this makes sense, such as a breaking ocean wave modeled by the shallow water equations, in most physical situations, this doesn't make sense. For instance, the density of a gas cannot be triple valued at a given point.

As seen in **Problem 1**, one way to determine the correct physical behavior can be achieved by adopting the vanishing discontinuity approach. There is yet another approach that results in a differential *integral* formulation that is often more convenient to work with. This approach is available by considering so-called the *weak solutions* and this is discussed in the next section in more detail.

2.3. Weak solutions

In order to successfully seek for physically meaningful solutions $u(x,t)$ of PDEs that are relevant to various physical phenomena, it would be much more desirable if we can relax those mathematical constraints on smoothness in $u(x,t)$. In other words, we wish to come up with a mathematical technique that can

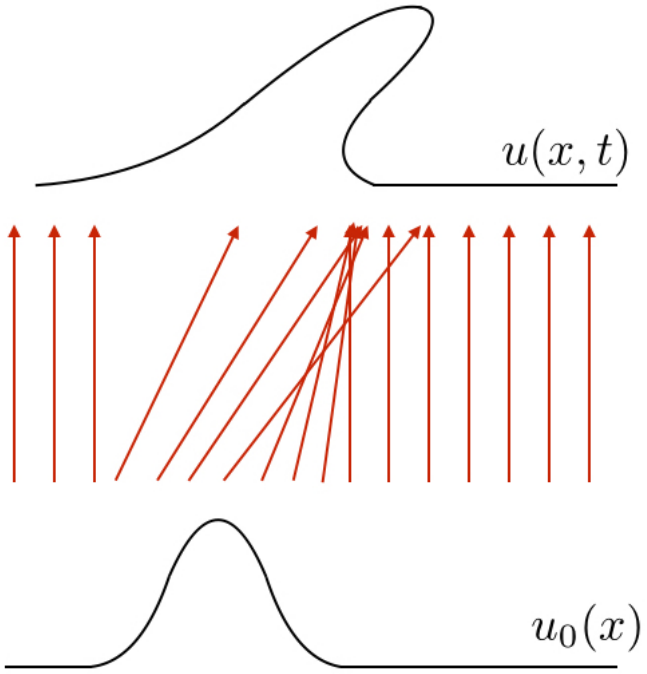


Figure 7. Triple-valued solution to Burgers' equation for large $t > t_b$.

be applied more generally to rewrite a differential equation in a form where less regularity is required to define a 'solution'. The weak solution approach is, in that sense, one such technique we are now considering. The basic idea is to take the PDE, multiply by a *smooth* "test function", integrate one or more times over some domain, and then use integration by parts to move derivatives off the function u and onto the smooth test function. The outcome is an equation involving fewer derivatives on u, and hence requiring less smoothness.

Definition: The function $u(x,t)$ is called a *weak solution* of the scalar conservation law $u_t + f_x = 0$ if it satisfies the following condition for all test functions $\phi(x,t) \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$:

$$\int_{\mathbb{R}^+} \int_{\mathbb{R}} [\phi_t u + \phi_x f(u)] dx dt = -\int_{\mathbb{R}} \phi(x,0) u(x,0) dx. \tag{2.31}$$

Note: C_0^1 is the space of functions that are continuously differentiable (C^1) with compact support.

Note: $f \in C_0(\mathbb{R})$ iff $f = 0$ in outside of some bounded sets and the support of f lies in a compact set. The support of f, $supp(f) = \{x \in X : f(x) \neq 0\}$.

Remark: One can obtain Eq. 2.31 by multiplying ϕ to $u_t + f_x = 0$ and then integrate over space and time,

$$\int_{\mathbb{R}^+} \int_{\mathbb{R}} [\phi u_t + \phi f(u)_x] dx dt = 0. \tag{2.32}$$

Finally, integrating Eq. 2.32 by part gives the definition of a weak solution in Eq. 2.31. Notice that nearly all the boundary terms which normally arise through integration by parts drop out because ϕ has compact support, hence becomes zero outside of some bounded region of the x-t plane. The RHS in Eq. 2.31 bears the initial conditions of the PDE which cannot be ignored in the weak formulation.

Quick summary: A nice feature of Eq. 2.31 is that the derivatives are on the smooth test function ϕ, and no longer on u and $f(u)$. This enables Eq. 2.31 to take some discontinuous u as a solution in this weak sense.

Remark: If u is a weak solution, then u also satisfies the original integral conservation law, and vice versa.

Remark: With the help of weak solutions, can we say we are now happy about solving nonlinear scalar conservation laws? The answer is not really yet, unfortunately. One of the reasons is that weak solutions are often not unique and therefore, we need some criteria to choose a physically correct weak solution among choices. To do this, we will consider a condition called the *'entropy condition'* at the end of this chapter.

2.4. The Riemann problem

The conservation law together with piecewise constant data separated by a single discontinuity is known as the Riemann problem (RP). There are two physically admissible types of solutions, (i) shock solution, and (ii) rarefaction solution, which we will consider here in detail.

Consider the general conservation laws written as

$$u_t + (f(u))_x = 0, \tag{2.33}$$

where $f(u)$ is convex. The RP involves a PDE with piecewise constant initial data,

$$u(x,0) = \begin{cases} u_l & \text{if } x < 0 \\ u_r & \text{if } x > 0 \end{cases} \tag{2.34}$$

and the form of the solution, as will be shown, closely depends on the relation between u_l and u_r.

- **Case I**: $u_l > u_r$ In this case, there is a unique weak solution

$$u(x,t) = \begin{cases} u_l & \text{if } x < st \\ u_r & \text{if } x > st \end{cases} \tag{2.35}$$

where s is a shock speed, the speed at which the discontinuity travels. We are going to study how to compute a general expression for the shock speed in the next section. The characteristics in each left and right regions where u is constant (i.e., either u_l or u_r) go *into* the shock as time advances. See Fig. 8 and Fig. 9.

Note: Since the flux is convex, $\lambda'(u) = f''(u) > 0$, λ is monotone increasing, hence $\lambda(u_l) > \lambda(u_r)$. We note for the RP with a shock solution, the characteristic speeds $f'(u)$ satisfy the following converging characteristic condition:

$$f'(u_l) > s > f'(u_r), \tag{2.36}$$

or equivalently,

$$\lambda(u_l) > s > \lambda(u_r), \tag{2.37}$$

where s is a shock speed.

- **Case II**: $u_l < u_r$ In this case there are infinitely many weak solutions, therefore, we need to choose a physically correct weak solution. Our first attempt is to apply the exact same idea as in **Case I** in which the discontinuity propagates with speed s. This now allows the characteristics *go out* of the shock as illustrated in Fig. 10. This type of weak solution is called the entropy violating solution and needs be rejected. One crucial reason for rejecting this solution as a physical solution is because the solution is not stable to perturbation (also recall the three requirements for well-posed PDEs we studied in Chapter 2). This means that small perturbations of the initial data lead to large changes in the solution. For example, if the data is smeared out little bit, or if a small amount of viscosity is added to the equation, the solution changes completely.

  Another weak solution is the rarefaction wave

$$u(x,t) = \begin{cases} u_l & \text{if } x < u_l t \\ x/t & \text{if } u_l t \le x \le u_r t \\ u_r & \text{if } x > u_r t \end{cases} \tag{2.38}$$

  This solution is stable to perturbation and is in fact the physically correct weak solution satisfying the vanishing viscosity approach.

  Note: Since the flux is convex, $\lambda'(u) = f''(u) > 0$, λ is monotone increasing, hence $\lambda(u_l) < \lambda(u_r)$. We note for the RP with a rarefaction solution, the characteristic speeds $f'(u)$ satisfy the following diverging characteristic condition:

$$f'(u_l) < f'(u_r), \tag{2.39}$$

  or equivalently,

$$\lambda(u_l) < \lambda(u_r). \tag{2.40}$$

Remark: Before proceeding to the next section, we briefly study four variants of the integral form of conservation laws $u_t + f_x = 0$. Recalled that we already

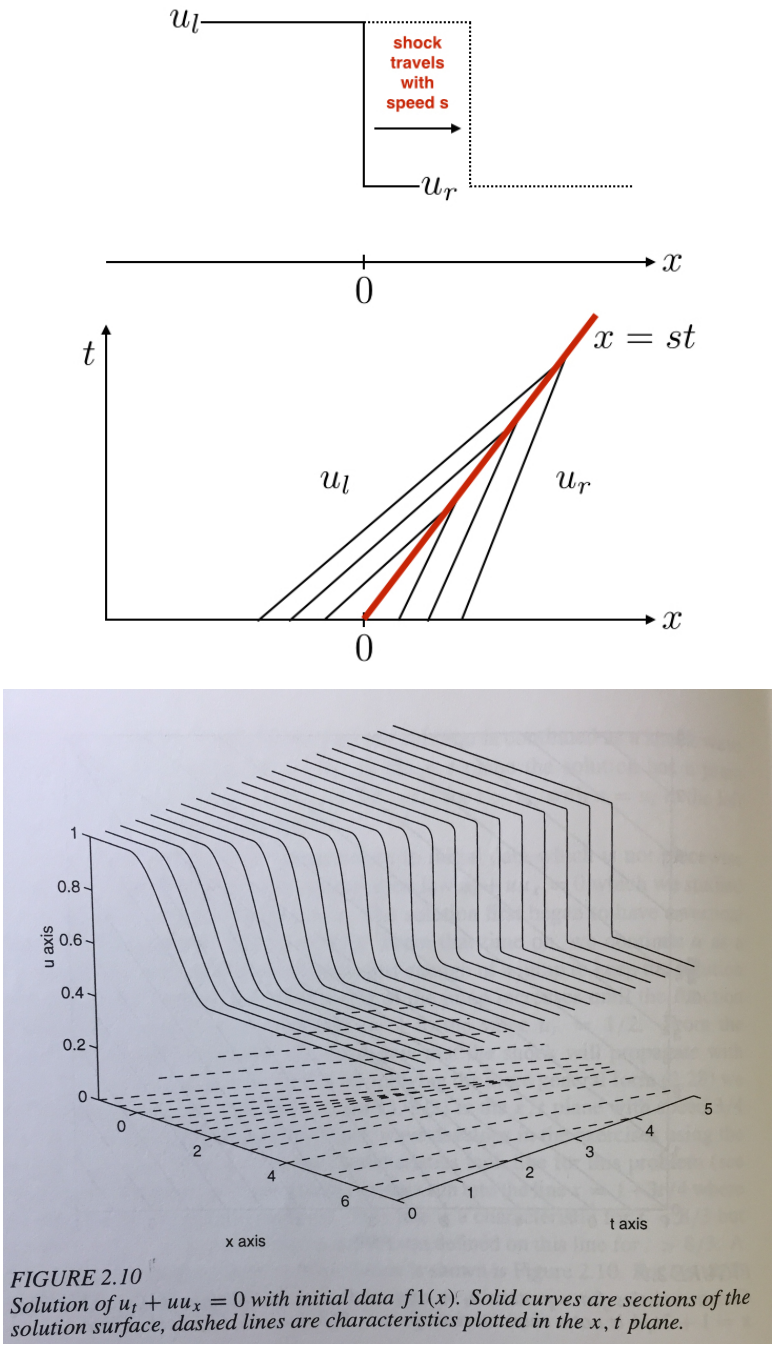


Figure 8. Weak solution of shock wave to the Riemann problem $u_l > u_r$. In addition, the last figure shows that a shock can be formed despite the initial condition is not discontinuous – Extracted from the book, "Introduction to Partial Differential Equations with MATLAB", by Jeffery M. Cooper.

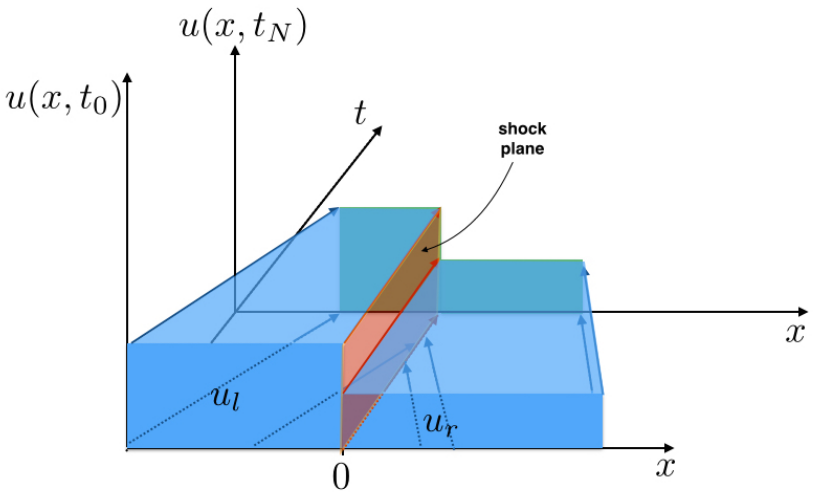


Figure 9. Weak solution of shock wave to the Riemann problem $u_l > u_r$. The characteristic curves are drawn in blue in the x-t plane. The dark orange shaded plane is the shock plane due to the crossing of the characteristics from the two discontinuous initial data u_l and u_r. The shock plane travels with the shock speed s which will be studied by considering the Rankine-Hugoniot jump condition in the next section.

have studied this in Chapter 1. This time, we choose a control volume $\mathcal{V} = [x_L, x_R] \times [t_1, t_2]$ on the x-t plane.

1. *Integral form I:*

$$\frac{d}{dt}\int_{x_L}^{x_R} u(x,t)dx = f(u(x_L,t)) - f(u(x_R,t)) \tag{2.41}$$

2. *Integral form II:* Integrating *Integral form I* in time gives

$$\int_{x_L}^{x_R} u(x,t_2)dx - \int_{x_L}^{x_R} u(x,t_1)dx = \int_{t_1}^{t_2} f(u(x_L,t))dt - \int_{t_1}^{t_2} f(u(x_R,t))dt \tag{2.42}$$

3. *Integral form III:* Integrating $u_t + f_x = 0$ in any domain $\mathcal{V}$ in the x-t plane and using Green's theorem, we obtain

$$\oint_{\partial\mathcal{V}} [udx - f(u)dt] = 0 \tag{2.43}$$

4. *Integral form IV:* The last variant is the integral relation that the weak or generalized solution u satisfies (see also Eq. 2.31) for all test function $\phi(x,t) \in C_0^1(\mathbb{R} \times \mathbb{R}^+)$:

$$\int_{\mathbb{R}^+}\int_{\mathbb{R}} [\phi_t u + \phi_x f(u)]dxdt = -\int_{\mathbb{R}} \phi(x,0)u(x,0)dx. \tag{2.44}$$

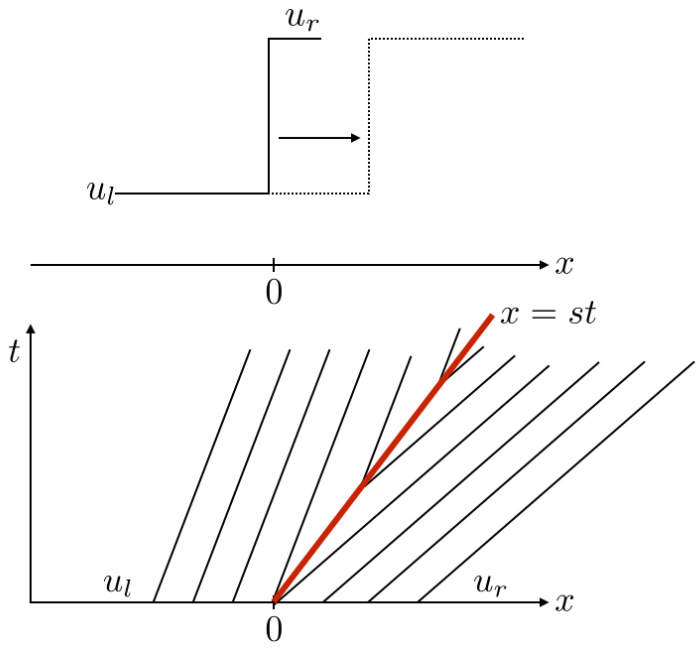


Figure 10. Entropy-violating shock that should be rejected.

2.5. Shock speed: the Rankine-Hugoniot jump condition

The propagating shock solution in Eq. 2.35 is a weak solution only with a proper value of the shock speed s. In fact, a correct shock speed s can be determined by considering conservation – called the Rankine-Hugoniot jump condition.

Consider a solution $u(x,t)$ such that $u(x,t)$ and $f(u)$ and their derivatives are continuous everywhere except on a line $S = S(t)$ on the x-t plane across which $u(x,t)$ has a jump discontinuity. Choose two fixed points x_L and x_R such that $x_L < S(t) < s_R$. Adopting *Integral form I* on the control volume $[x_L, x_R]$, we have

$$f(u(x_L,t)) - f(u(x_R,t)) = \frac{d}{dt}\int_{x_L}^{S(t)} u(x,t)dx + \frac{d}{dt}\int_{S(t)}^{x_R} u(x,t)dx, \qquad (2.45)$$

which becomes

$$f(u(x_L,t))-f(u(x_R,t)) = \Big(u(S_L,t)-u(S_R,t)\Big)\frac{dS}{dt}+\int_{x_L}^{S(t)} u_t(x,t)dx+\int_{S(t)}^{x_R} u_t(x,t)dx, \qquad (2.46)$$

where

$$u(S_L,t) = \lim_{x\uparrow S(t)} u(S(t),t), \qquad (2.47)$$

$$u(S_R,t) = \lim_{x\downarrow S(t)} u(S(t),t) \qquad (2.48)$$

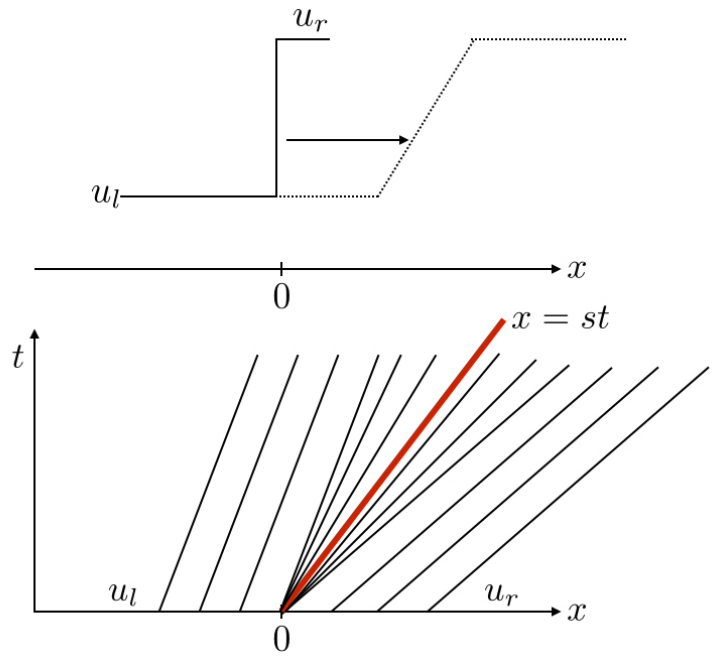


Figure 11. Entropy satisfying weak solution – the rarefaction wave.

Note the two integrals in Eq. 2.46 become

$$\int_{x_L}^{S(t)} u_t(x,t)dx = -\int_{x_L}^{S(t)} f_x(u(x,t))dx = f(u(x_L,t)) - f(u(S_L,t)), \quad (2.49)$$

$$\int_{S(t)}^{x_R} u_t(x,t)dx = -\int_{S(t)}^{x_R} f_x(u(x,t))dx = f(u(S_R,t)) - f(u(x_R,t)). \quad (2.50)$$

After canceling $f(u(x_L,t)) - f(u(x_R,t))$ from both sides, we finally obtain

$$f(u(S_L,t)) - f(u(S_R,t)) = \Big(u(S_L,t) - u(S_R,t)\Big)s, \quad (2.51)$$

where we introduced $s = dS/dt$ the speed of the discontinuity.

Definition: The relation in Eq. 2.51 is called the *Rankine-Hugoniot jump condition* (RH condition) and it provides a relation between the shock speed s and the states $u_l = u(S_L,t)$ and $u_r = u(S_R,t)$. We often denote shock speed s in the RH condition using brackets as follow:

$$s = \frac{[f]}{[u]} \equiv \frac{\lim_{x\downarrow S(t)} f(u(x,t),t) - \lim_{x\uparrow S(t)} f(u(x,t),t)}{\lim_{x\downarrow S(t)} u(x,t) - \lim_{x\uparrow S(t)} u(x,t)}. \quad (2.52)$$

Problem 3 Consider Burgers' equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \quad (2.53)$$

(a) By multiplying the equation by $2u$, show that you can derive a new conservation law for u^2. What is the new flux function?
(b) Show that the original Burgers' equation and the new derived equation have different weak solutions (Hint: It suffices to show that there exist two different shock speeds from the two equations for the Riemann problem with $u_l > u_r$.).

2.6. Entropy conditions

As demonstrated in **Problem 3** above, there are situations in which the weak solution is not unique. It is therefore natural to ask for an additional condition to pick out the physically relevant solution. Recall that we've already seen there is an obvious condition for the characteristic speeds in Eq. 2.36. A shock should have characteristics *going into* the shock as time evolves. We are now ready to state it and call it the entropy condition:

Definition: A discontinuity propagating with speed s given by Eq. 2.51 (or equivalently, Eq. 2.52) – that is, the two data states u_l and u_r are connected through a single discontinuity with its speed s – satisfies the *entropy condition* if

$$f'(u_l) > s > f'(u_r), \tag{2.54}$$

or equivalently,

$$\lambda(u_l) > s > \lambda(u_r). \tag{2.55}$$

Remark: On the other hand, if the two data states u_l and u_r are connected through a smooth transition – i.e., rarefaction wave – the divergence relation of the characteristics holds:

$$f'(u_l) < f'(u_r) \tag{2.56}$$

or equivalently,

$$\lambda(u_l) < \lambda(u_r). \tag{2.57}$$

Example: Let's consider Burgers' equation on $\mathbb{R}$ with the following initial conditions:

$$u(x,0) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases} \tag{2.58}$$

We can first try to obtain an entropy violating solution, and we know that this solution needs to be rejected anyway as it is ill-posed. But we are going to find this solution to practice what we already learned in this chapter. If we apply the RH condition to this problem – which *is wrong to do so* – to compute the shock speed s, we get

$$s = \frac{f(u_r) - f(u_l)}{u_r - u_l} = \frac{1}{2}. \tag{2.59}$$

This results in the following entropy violating self-similar solution

$$u(x,t) = \begin{cases} 0 & \text{if } \frac{x}{t} < \frac{1}{2} \\ 1 & \text{if } \frac{x}{t} > \frac{1}{2} \end{cases} \tag{2.60}$$

which is shown in Fig.12.

Let us try again to get the correct weak solution this time. Consider the following self-similar solution

$$u(x,t) = \begin{cases} 0 & \text{if } \frac{x}{t} < 0 \\ x/t & \text{if } 0 < \frac{x}{t} < 1 \\ 1 & \text{if } \frac{x}{t} > 1. \end{cases} \tag{2.61}$$

We can check that the wave diagram for this solution is plotted in Fig. 13. It is also easy to check if this solution, especially the part in the expansion region, satisfies the Burgers' equation. To see this,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \frac{\partial}{\partial t}\Big(\frac{x}{t}\Big) + \frac{x}{t}\frac{\partial}{\partial x}\Big(\frac{x}{t}\Big) = -\frac{x}{t^2} + \frac{x}{t}\frac{1}{t} = 0. \tag{2.62}$$

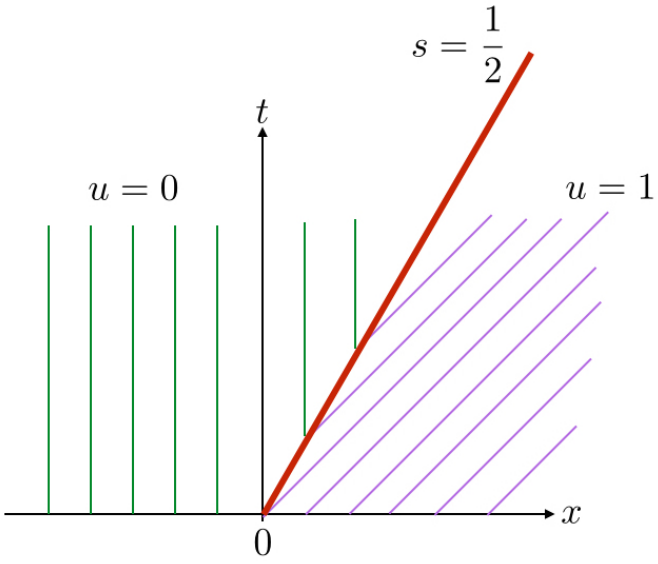


Figure 12. Wave diagram for the wrong entropy violating weak solution.

Example: Let's consider Burgers' equation on $\mathbb{R}$ with the following initial conditions for $t \leq 4/3$:

$$u(x,0) = \begin{cases} 1 & \text{if } |x| < 1/3 \\ 0 & \text{if } |x| > 1/3 \end{cases} \tag{2.63}$$

We see that the jump at $x = -1/3$ creates a rarefaction wave solution; the jump at $x = 1/3$ crates a shock solution. For $t \leq 4/3$ the shock and the rarefaction fan do not intersect each other and therefore, we can seek for the exact piecewise-linear solution as follows.

Let us first compute the shock speed using RH with $u_l = 1$ and $u_r = 0$:

$$s = \frac{f(u_r) - f(u_l)}{u_r - u_l} = \frac{1}{2}, \tag{2.64}$$

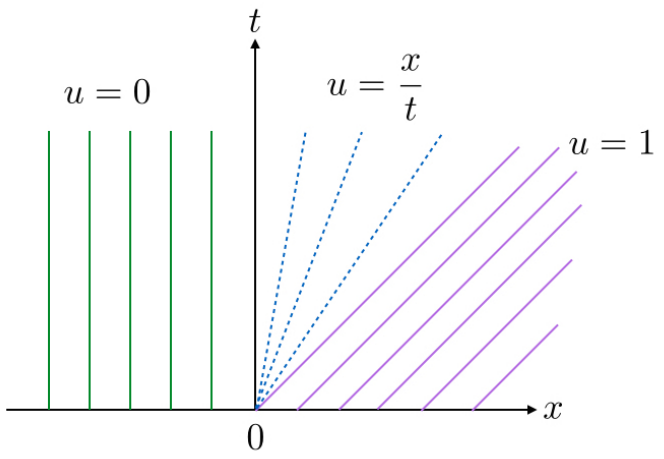


Figure 13. Wave diagram for the correct rarefaction wave weak solution.

which gives the characteristic curve (the red thick line in Fig. 14) for shock $x - 1/2t = 1/3$.
We also consider the first characteristic curve right next to the rarefaction region – this is the left most purple line in Fig. 14. Since the characteristic slope is $f'(u)|_{u=1} = 1$, we obtain the relation $x - t = -1/3$. From these we easily obtain the exact weak solution as follows:

$$u(x,t) = \begin{cases} 0 & \text{if } x < -\frac{1}{3} \\ \frac{x+1/3}{t} & \text{if } -\frac{1}{3} < x < t - \frac{1}{3} \\ 1 & \text{if } t - \frac{1}{3} < x < \frac{1}{2}t + \frac{1}{3} \\ 0 & \text{if } x > \frac{1}{2}t + \frac{1}{3} \end{cases} \tag{2.65}$$

Note that at $t = \frac{4}{3}$ we get $\frac{1}{2}t + \frac{1}{3} = t - \frac{1}{3}$, and as a result, the shock and the rarefaction solutions intersect for $t > \frac{4}{3}$.

Problem 4 Solve Burgers' equation on $\mathbb{R}$ for small enough $t \le t_b$ that allows the exact piecewise-linear weak solution with the following initial conditions:

$$u(x,0) = \begin{cases} 2 & \text{if } |x| < 1/2 \\ -1 & \text{if } |x| > 1/2 \end{cases} \tag{2.66}$$

Find the time t_b when the two waves first intersect. Draw a wave diagram for the weak solution.

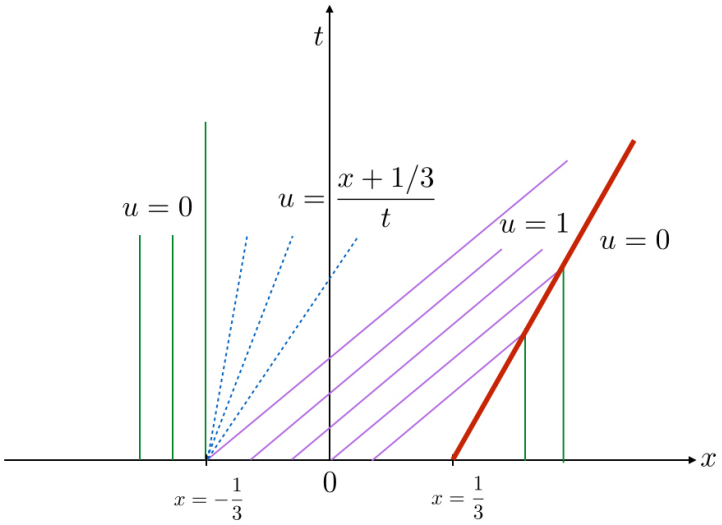


Figure 14. Wave diagram of the weak solution for $t \leq 4/3$.

Chapter 3

Discrete Numerical Approaches

We review several key ideas on numerical methods that discretize PDEs and provide approximated solutions to numerical PDE models derived from the analytical PDEs. Three major methods are briefly described along with the principal advantages and drawbacks in each method. Three solution schemes include:

- Finite difference method (FD)
- Finite volume method (FV)
- Finite element method (FE)

Some other approaches also used in many CFD applications include:

- Discontinuous Galerkin (DG) (or, discontinuous FE as compared to the standard 'continuous' FE)
- Spectral element (SE)

In general, a proper choice of numerical approaches strongly depends on various components of your problem, including especially the following factors:

- Flow regimes – e.g., compressible (FV) vs. incompressible (FD), high Mach number (FV) vs. low Mach number (low Mach number scheme), turbulent (subgrid models) vs. laminar (boundary layer), advection dominated (FV, FD, DG) vs. diffusion dominated (FE)
- Physics of flows – e.g., macroscopic (fluid models: FV, FD, FE) vs. microscopic (kinetic models: PIC – particle-in-cell), hydrodynamcis vs. magnetohydrodynamics vs. rad-hydro, single-fluid (sigle bulk velocity) vs. multi-fluid (multiple bulk velocity), advection dominated (explicit) vs. diffusion dominated (implicit) vs. combined (explicit & implicit via operator split), gravitational flow (elliptic solver)
- Geometry of flows – e.g., rectangular domain (FD) vs. engineering flow (complicated physical boundaries such as bridges, airplane, airfoils, cars, buildings – mostly FE, but also FV), localized dynamics (AMR – adaptive mesh refinements; stretched grid) vs. global dynamics (UG – uniform grid)

- Numerical issues – ease of high-order implementation (FD, FE, DG) vs. difficulty in high-order implementation (FV), ease of multi-dimensional extension (FD) vs. difficulty in multi-dimensional extension (FV)

Our primary interest in this course lies in studying the first two methods, FD and FV. Later, we are going to use FD and FV approaches to solve linear advection equation and linear hyperbolic systems. Such fundamental ideas of solving linear hyperbolic PDEs will be extended to the nonlinear cases.

For the rest of the study in this chapter, we are going to refer to a short article by Joaquim Peiró and Spencer Sherwin which provides a nice overview and comparison of three discrete finite approaches, FD, FE, and FV:

- "Finite difference, finite element and finite volume methods for partial differential equations" (enclosed below).

8.2

FINITE DIFFERENCE, FINITE ELEMENT AND FINITE VOLUME METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS

Joaquim Peiró and Spencer Sherwin
Department of Aeronautics, Imperial College, London, UK

There are three important steps in the computational modelling of any physical process: (i) problem definition, (ii) mathematical model, and (iii) computer simulation.

The first natural step is to define an idealization of our problem of interest in terms of a set of relevant quantities which we would like to measure. In defining this idealization we expect to obtain a well-posed problem, this is one that has a unique solution for a given set of parameters. It might not always be possible to guarantee the fidelity of the idealization since, in some instances, the physical process is not totally understood. An example is the complex environment within a nuclear reactor where obtaining measurements is difficult.

The second step of the modeling process is to represent our idealization of the physical reality by a mathematical model: the governing equations of the problem. These are available for many physical phenomena. For example, in fluid dynamics the Navier–Stokes equations are considered to be an accurate representation of the fluid motion. Analogously, the equations of elasticity in structural mechanics govern the deformation of a solid object due to applied external forces. These are complex general equations that are very difficult to solve both analytically and computationally. Therefore, we need to introduce simplifying assumptions to reduce the complexity of the mathematical model and make it amenable to either exact or numerical solution. For example, the irrotational (without vorticity) flow of an incompressible fluid is accurately represented by the Navier–Stokes equations but, if the effects of fluid viscosity are small, then Laplace's equation of *potential flow* is a far more efficient description of the problem.

S. Yip (ed.),
Handbook of Materials Modeling. Volume I: Methods and Models, 1–32.

After the selection of an appropriate mathematical model, together with suitable boundary and initial conditions, we can proceed to its solution. In this chapter we will consider the numerical solution of mathematical problems which are described by partial differential equations (PDEs). The three classical choices for the numerical solution of PDEs are the finite difference method (FDM), the finite element method (FEM) and the finite volume method (FVM).

The FDM is the oldest and is based upon the application of a local Taylor expansion to approximate the differential equations. The FDM uses a topologically square network of lines to construct the discretization of the PDE. This is a potential bottleneck of the method when handling complex geometries in multiple dimensions. This issue motivated the use of an integral form of the PDEs and subsequently the development of the finite element and finite volume techniques.

To provide a short introduction to these techniques we shall consider each type of discretization as applied to one-dimensional PDEs. This will not allow us to illustrate the geometric flexibility of the FEM and the FVM to their full extent, but we will be able to demonstrate some of the similarities between the methods and thereby highlight some of the relative advantages and disadvantages of each approach. For a more detailed understanding of the approaches we refer the reader to the section on suggested reading at the end of the chapter.

The section is structured as follows. We start by introducing the concept of conservation laws and their differential representation as PDEs and the alternative integral forms. We next discusses the classification of partial differential equations: elliptic, parabolic, and hyperbolic. This classification is important since the type of PDE dictates the form of boundary and initial conditions required for the problem to be well-posed. It also, permits in some cases, e.g., in hyperbolic equations, to identify suitable schemes to discretise the differential operators. The three types of discretisation: FDM, FEM and FVM are then discussed and applied to different types of PDEs. We then end our overview by discussing the numerical difficulties which can arise in the numerical solution of the different types of PDEs using the FDM and provides an introduction to the assessment of the stability of numerical schemes using a Fourier or Von Neumann analysis.

Finally we note that, given the scientific background of the authors, the presentation has a bias towards fluid dynamics. However, we stress that the fundamental concepts presented in this chapter are generally applicable to continuum mechanics, both solids and fluids.

1. Conservation Laws: Integral and Differential Forms

The governing equations of continuum mechanics representing the kinematic and mechanical behaviour of general bodies are commonly referred

to as *conservation laws*. These are derived by invoking the conservation of mass and energy and the momentum equation (Newton's law). Whilst they are equally applicable to solids and fluids, their differing behaviour is accounted for through the use of a different constitutive equation.

The general principle behind the derivation of conservation laws is that the rate of change of $u(\boldsymbol{x}, t)$ within a volume V plus the flux of u through the boundary A is equal to the rate of production of u denoted by $S(u, \boldsymbol{x}, t)$. This can be written as

$$\frac{\partial}{\partial t}\int_V u(\boldsymbol{x}, t)\,\mathrm{d}V + \int_A \boldsymbol{f}(u)\cdot\boldsymbol{n}\,\mathrm{d}A - \int_V S(u, \boldsymbol{x}, t)\,\mathrm{d}V = 0 \tag{1}$$

which is referred to as the *integral* form of the conservation law. For a fixed (independent of t) volume and, under suitable conditions of smoothness of the intervening quantities, we can apply Gauss' theorem

$$\int_V \nabla\cdot\boldsymbol{f}\,\mathrm{d}V = \int_A \boldsymbol{f}\cdot\boldsymbol{n}\,\mathrm{d}A$$

to obtain

$$\int_V \left(\frac{\partial u}{\partial t} + \nabla\cdot\boldsymbol{f}(u) - S\right)\mathrm{d}V = 0. \tag{2}$$

For the integral expression to be zero for any volume V, the integrand must be zero. This results in the *strong* or differential form of the equation

$$\frac{\partial u}{\partial t} + \nabla\cdot\boldsymbol{f}(u) - S = 0. \tag{3}$$

An alternative *integral* form can be obtained by the method of weighted residuals. Multiplying Eq. (3) by a *weight* function $w(\boldsymbol{x})$ and integrating over the volume V we obtain

$$\int_V \left(\frac{\partial u}{\partial t} + \nabla\cdot\boldsymbol{f}(u) - S\right) w(\boldsymbol{x})\,\mathrm{d}V = 0. \tag{4}$$

If Eq. (4) is satisfied for any weight function $w(\boldsymbol{x})$, then Eq. (4) is equivalent to the differential form (3). The smoothness requirements on $\boldsymbol{f}$ can be relaxed by applying the Gauss' theorem to Eq. (4) to obtain

$$\int_V \left[\left(\frac{\partial u}{\partial t} - S\right) w(\boldsymbol{x}) - \boldsymbol{f}(u)\cdot\nabla w(\boldsymbol{x})\right]\mathrm{d}V + \int_A \boldsymbol{f}\cdot\boldsymbol{n}\, w(\boldsymbol{x})\,\mathrm{d}A = 0. \tag{5}$$

This is known as the *weak* form of the conservation law.

Although the above formulation is more commonly used in fluid mechanics, similar formulations are also applied in structural mechanics. For instance, the well-known principle of virtual work for the static equilibrium of a body [1], is given by

$$\delta W = \int_V (\nabla \boldsymbol{\sigma} + \boldsymbol{f}) \cdot \delta \boldsymbol{v} \, \mathrm{d}V = 0$$

where δW denotes the virtual work done by an arbitrary virtual velocity $\delta \boldsymbol{v}$, $\boldsymbol{\sigma}$ is the stress tensor and $\boldsymbol{f}$ denotes the body force. The similarity with the method of weighted residuals (4) is evident.

2. Model Equations and their Classification

In the following we will restrict ourselves to the analysis of one-dimensional conservation laws representing the transport of a scalar variable $u(x,t)$ defined in the domain $\Omega = \{x, t : 0 \leq x \leq 1, 0 \leq t \leq T\}$. The convection–diffusion-reaction equation is given by

$$\mathcal{L}(u) = \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(au - b\frac{\partial u}{\partial x}\right) - r\,u = s \tag{6}$$

together with appropriate boundary conditions at $x = 0$ and 1 to make the problem well-posed. In the above equation $\mathcal{L}(u)$ simply represents a linear differential operator. This equation can be recast in the form (3) with $f(u) = au - \partial u/\partial x$ and $S(u) = s + ru$. It is linear if the coefficient a, b, r and s are functions of x and t, and non-linear if any of them depends on the solution, u.

In what follows, we will use for convenience the convention that the presence of a subscript x or t under a expression indicates a derivative or partial derivative with respect to this variable, for example

$$u_x(x) = \frac{\mathrm{d}u}{\mathrm{d}x}(x); \qquad u_t(x,t) = \frac{\partial u}{\partial t}(x,t); \qquad u_{xx}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t).$$

Using this notation, Eq. (6) is re-written as

$$u_t + (au - bu_x)_x - ru = s.$$

2.1. Elliptic Equations

The steady-state solution of Eq. (6) when advection and source terms are neglected, i.e., a=0 and s=0, is a function of x only and satisfies the Helmholtz equation

$$(bu_x)_x + ru = 0. \tag{7}$$

This equation is elliptic and its solution depends on two families of integration constants that are fixed by prescribing boundary conditions at the ends of the domain. One can either prescribe Dirichlet boundary conditions at both ends, e.g., $u(0) = \alpha_0$ and $u(1) = \alpha_1$, or substitute one of them (or both if $r \neq 0$) by a Neumann boundary condition, e.g., $u_x(0) = g$. Here α_0, α_1 and g are known constant values. We note that if we introduce a perturbation into a Dirichlet boundary condition, e.g., $u(0) = \alpha_0 + \epsilon$, we will observe an instantaneous modification to the solution throughout the domain. This is indicative of the elliptic nature of the problem.

2.2. Parabolic Equations

Taking $a = 0$, $r = 0$ and $s = 0$ in our model, Eq. (6) leads to the heat or diffusion equation

$$u_t - (b\,u_x)_x = 0, \tag{8}$$

which is parabolic. In addition to appropriate boundary conditions of the form used for elliptic equations, we also require an initial condition at $t = 0$ of the form $u(x, 0) = u_0(x)$ where u_0 is a given function.

If b is constant, this equation admits solutions of the form $u(x, t) = Ae^{\beta t} \sin kx$ if $\beta + k^2 b = 0$. A notable feature of the solution is that it decays when b is positive as the exponent $\beta < 0$. The rate of decay is a function of b. The more diffusive the equation (i.e., larger b) the faster the decay of the solution is. In general the solution can be made up of many sine waves of different frequencies, i.e., a Fourier expansion of the form

$$u(x, t) = \sum_m A_m \mathrm{e}^{\beta_m t} \sin\, k_m x,$$

where A_m and k_m represent the amplitude and the frequency of a Fourier mode, respectively. The decay of the solution depends on the Fourier contents of the initial data since $\beta_m = -k_m^2 b$. High frequencies decay at a faster rate than the low frequencies which physically means that the solution is being smoothed. This is illustrated in Fig. 1 which shows the time evolution of $u(x, t)$ for an initial condition $u_0(x) = 20x$ for $0 \leq x \leq 1/2$ and $u_0(x) = 20(1 - x)$ for $1/2 \leq x \leq 1$. The solution shows a rapid smoothing of the slope discontinuity of the initial condition at $x = 1/2$. The presence of a positive diffusion ($b > 0$) physically results in a smoothing of the solution which stabilizes it. On the other hand, negative diffusion ($b < 0$) is de-stabilizing but most physical problems have positive diffusion.

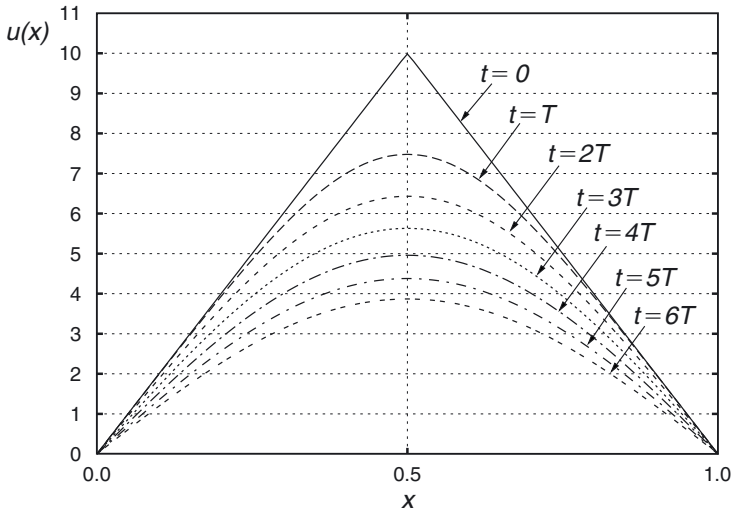


Figure 1. Rate of decay of the solution to the diffusion equation.

2.3. Hyperbolic Equations

A classic example of hyperbolic equation is the linear advection equation

$$u_t + a\,u_x = 0, \tag{9}$$

where a represents a constant velocity. The above equation is also clearly equivalent to Eq. (6) with $b = r = s = 0$. This hyperbolic equation also requires an initial condition, $u(x, 0) = u_0(x)$. The question of what boundary conditions are appropriate for this equation can be more easily be answered after considering its solution. It is easy to verify by substitution in (9) that the solution is given by $u(x, t) = u_0(x - at)$. This describes the propagation of the quantity $u(x, t)$ moving with speed "a" in the x-direction as depicted in Fig. 2. The solution is constant along the *characteristic line* $x - at = c$ with $u(x, t) = u_0(c)$.

From the knowledge of the solution, we can appreciate that for $a > 0$ a boundary condition should be prescribed at $x = 0$, (e.g., $u(0) = \alpha_0$) where information is being fed into the solution domain. The value of the solution at $x = 1$ is determined by the initial conditions or the boundary condition at $x = 0$ and cannot, therefore, be prescribed. This simple argument shows that, in a hyperbolic problem, the selection of appropriate conditions at a boundary point depends on the solution at that point. If the velocity is negative, the previous treatment of the boundary conditions is reversed.

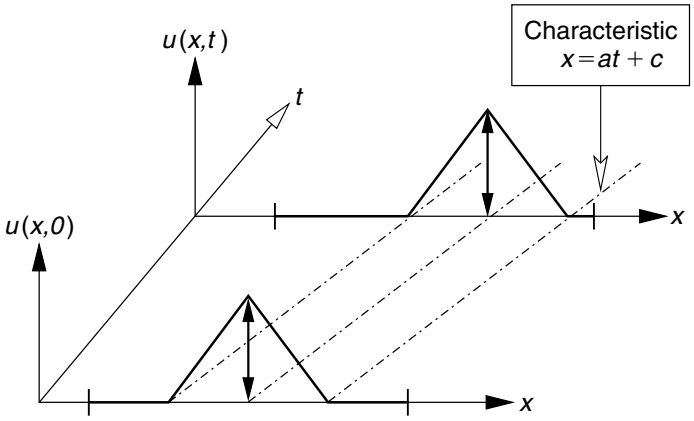


Figure 2. Solution of the linear advection equation.

The propagation velocity can also be a function of space, i.e., $a = a(x)$ or even the same as the quantity being propagated, i.e., $a = u(x, t)$. The choice $a = u(x, t)$ leads to the non-linear inviscid Burgers' equation

$$u_t + u\, u_x = 0. \tag{10}$$

An analogous analysis to that used for the advection equation shows that $u(x, t)$ is constant if we are moving with a local velocity also given by $u(x, t)$. This means that some regions of the solution advance faster than other regions leading to the formation of sharp gradients. This is illustrated in Fig. 3. The initial velocity is represented by a triangular "zig-zag" wave. Peaks and troughs in the solution will advance, in opposite directions, with maximum speed. This will eventually lead to an overlap as depicted by the dotted line in Fig. 3. This results in a non-uniqueness of the solution which is obviously non-physical and to resolve this problem we must allow for the formation and propagation of discontinuities when two characteristics intersect (see Ref. [2] for further details).

3. Numerical Schemes

There are many situations where obtaining an exact solution of a PDE is not possible and we have to resort to approximations in which the infinite set of values in the continuous solution is represented by a finite set of values referred to as the *discrete* solution.

For simplicity we consider first the case of a function of one variable $u(x)$. Given a set of points x_i; $i = 1, \ldots, N$ in the domain of definition of $u(x)$, as

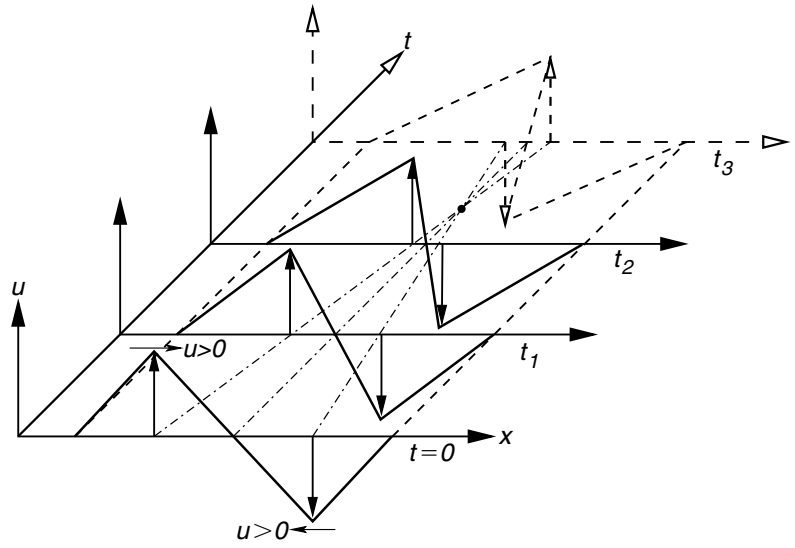


Figure 3. Formation of discontinuities in the Burgers' equation.

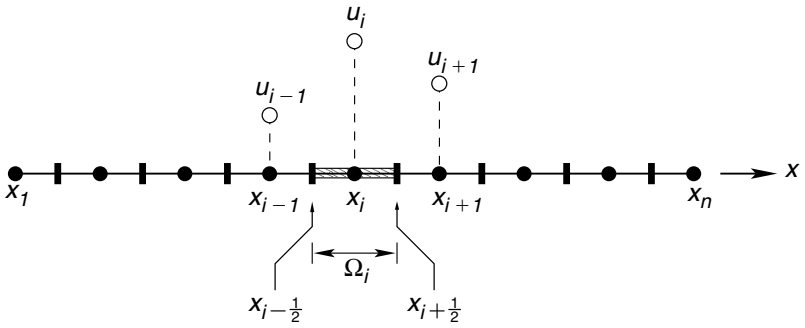


Figure 4. Discretization of the domain.

shown in Fig. 4, the numerical solution that we are seeking is represented by a discrete set of function values $\{u_1, \ldots, u_N\}$ that approximate u at these points, i.e., $u_i \approx u(x_i)$; $i = 1, \ldots, N$.

In what follows, and unless otherwise stated, we will assume that the points are equally spaced along the domain with a constant distance $\Delta x = x_{i+1} - x_i$; $i = 1, \ldots, N-1$. This way we will write $u_{i+1} \approx u(x_{i+1}) = u(x_i + \Delta x)$. This partition of the domain into smaller subdomains is referred to as a *mesh* or *grid*.

3.1. The Finite Difference Method (FDM)

This method is used to obtain numerical approximations of PDEs written in the strong form (3). The derivative of $u(x)$ with respect to x can be defined as

$$\begin{aligned} u_x|_i = u_x(x_i) &= \lim_{\Delta x \to 0} \frac{u(x_i + \Delta x) - u(x_i)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{u(x_i) - u(x_i - \Delta x)}{\Delta x} \\ &= \lim_{\Delta x \to 0} \frac{u(x_i + \Delta x) - u(x_i - \Delta x)}{2\Delta x}. \end{aligned} \tag{11}$$

All these expressions are mathematically equivalent, i.e., the approximation converges to the derivative as $\Delta x \to 0$. If Δx is small but finite, the various terms in Eq. (11) can be used to obtain approximations of the derivate u_x of the form

$$u_x|_i \approx \frac{u_{i+1} - u_i}{\Delta x} \tag{12}$$

$$u_x|_i \approx \frac{u_i - u_{i-1}}{\Delta x} \tag{13}$$

$$u_x|_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}. \tag{14}$$

The expressions (12)–(14) are referred to as forward, backward and centred finite difference approximations of $u_x|_i$, respectively. Obviously these approximations of the derivative are different.

3.1.1. Errors in the FDM

The analysis of these approximations is performed by using Taylor expansions around the point x_i. For instance an approximation to u_{i+1} using $n+1$ terms of a Taylor expansion around x_i is given by

$$\begin{aligned} u_{i+1} = u_i + u_x|_i\, \Delta x + u_{xx}|_i\, \frac{\Delta x^2}{2} + \cdots + \left.\frac{\mathrm{d}^n u}{\mathrm{d}x^n}\right|_i \frac{\Delta x^n}{n!} \\ + \underline{\frac{\mathrm{d}^{n+1} u}{\mathrm{d}x^{n+1}}(x^*)\frac{\Delta x^{n+1}}{(n+1)!}}. \end{aligned} \tag{15}$$

The underlined term is called the remainder with $x_i \le x^* \le x_{i+1}$, and represents the error in the approximation if only the first n terms in the expansion are kept. Although the expression (15) is exact, the position x^* is unknown.

To illustrate how this can be used to analyse finite difference approximations, consider the case of the forward difference approximation (12) and use the expansion (15) with $n = 1$ (two terms) to obtain

$$\frac{u_{i+1} - u_i}{\Delta x} = u_x|_i + \frac{\Delta x}{2} u_{xx}(x^*). \tag{16}$$

We can now write the approximation of the derivative as

$$u_x|_i = \frac{u_{i+1} - u_i}{\Delta x} + \epsilon_{\mathrm{T}} \tag{17}$$

where ϵ_{T} is given by

$$\epsilon_{\mathrm{T}} = -\frac{\Delta x}{2} u_{xx}(x^*). \tag{18}$$

The term ϵ_{T} is referred to as the *truncation error* and is defined as the difference between the exact value and its numerical approximation. This term depends on Δx but also on u and its derivatives. For instance, if $u(x)$ is a linear function then the finite difference approximation is exact and $\epsilon_{\mathrm{T}} = 0$ since the second derivative is zero in (18).

The *order* of a finite difference approximation is defined as the power p such that $\lim_{\Delta x \to 0}(\epsilon_{\mathrm{T}}/\Delta x^p) = \gamma \neq 0$, where γ is a finite value. This is often written as $\epsilon_{\mathrm{T}} = O(\Delta x^p)$. For instance, for the forward difference approximation (12), we have $\epsilon_{\mathrm{T}} = O(\Delta x)$ and it is said to be first-order accurate ($p = 1$).

If we apply this method to the backward and centred finite difference approximations (13) and (14), respectively, we find that, for constant Δx, their errors are

$$u_x|_i = \frac{u_i - u_{i-1}}{\Delta x} + \frac{\Delta x}{2} u_{xx}(x^*) \Rightarrow \epsilon_{\mathrm{T}} = O(\Delta x) \tag{19}$$

$$u_x|_i = \frac{u_{i+1} - u_{i-1}}{2\Delta x} - \frac{\Delta x^2}{12} u_{xxx}(x^\star) \Rightarrow \epsilon_{\mathrm{T}} = O(\Delta x^2) \tag{20}$$

with $x_{i-1} \leq x^* \leq x_i$ and $x_{i-1} \leq x^\star \leq x_{i+1}$ for Eqs. (19) and (20), respectively.

This analysis is confirmed by the numerical results presented in Fig. 5 that displays, in logarithmic axes, the exact and truncation errors against Δx for the backward and the centred finite differences. Their respective truncation errors ϵ_{T} are given by (19) and (20) calculated here, for lack of a better value, with $x^* = x^\star = x_i$. The exact error is calculated as the difference between the exact value of the derivative and its finite difference approximation.

The slope of the lines are consistent with the order of the truncation error, i.e., 1:1 for the backward difference and 1:2 for the centred difference. The discrepancies between the exact and the numerical results for the smallest values of Δx are due to the use of finite precision computer arithmetic or round-off error. This issue and its implications are discussed in more detail in numerical analysis textbooks as in Ref. [3].

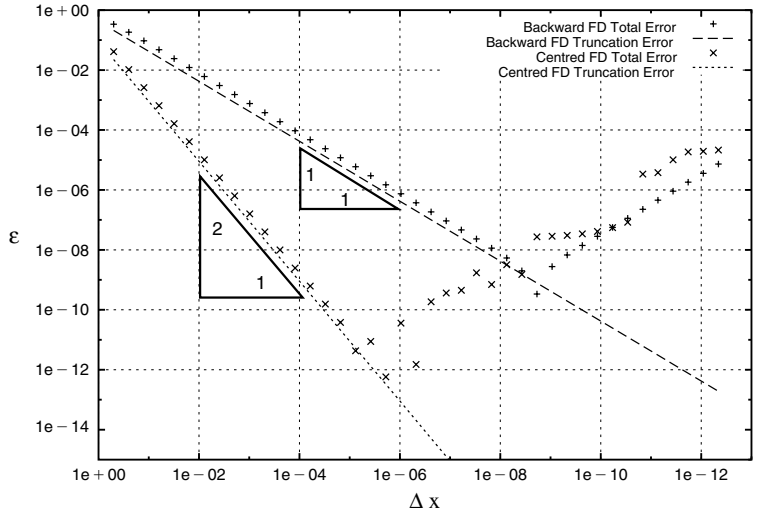


Figure 5. Truncation and rounding errors in the finite difference approximation of derivatives.

3.1.2. Derivation of approximations using Taylor expansions

The procedure described in the previous section can be easily transformed into a general method for deriving finite difference schemes. In general, we can obtain approximations to higher order derivatives by selecting an appropriate number of interpolation points that permits us to eliminate the highest term of the truncation error from the Taylor expansions. We will illustrate this with some examples. A more general description of this derivation can be found in Hirsch (1988).

A second-order accurate finite difference approximation of the derivative at x_i can be derived by considering the values of u at three points: x_{i-1}, x_i and x_{i+1}. The approximation is constructed as a weighted average of these values $\{u_{i-1}, u_i, u_{i+1}\}$ such as

$$u_x|_i \approx \frac{\alpha u_{i+1} + \beta u_i + \gamma u_{i-1}}{\Delta x}. \tag{21}$$

Using Taylor expansions around x_i we can write

$$u_{i+1} = u_i + \Delta x\, u_x|_i + \frac{\Delta x^2}{2}\, u_{xx}|_i + \frac{\Delta x^3}{6}\, u_{xxx}|_i + \cdots \tag{22}$$

$$u_{i-1} = u_i - \Delta x\, u_x|_i + \frac{\Delta x^2}{2}\, u_{xx}|_i - \frac{\Delta x^3}{6}\, u_{xxx}|_i + \cdots \tag{23}$$

Putting (22), (23) into (21) we get

$$\begin{aligned}\frac{\alpha u_{i+1}+\beta u_i+\gamma u_{i-1}}{\Delta x} &= (\alpha+\beta+\gamma)\frac{1}{\Delta x}u_i+(\alpha-\gamma)\,u_x|_i \\ &\quad +(\alpha+\gamma)\frac{\Delta x}{2}\,u_{xx}|_i+(\alpha-\gamma)\frac{\Delta x^2}{6}\,u_{xxx}|_i \\ &\quad +(\alpha+\gamma)\frac{\Delta x^3}{12}\,u_{xxxx}|_i+O(\Delta x^4) \qquad (24)\end{aligned}$$

We require three independent conditions to calculate the three unknowns α, β and γ. To determine these we impose that the expression (24) is consistent with increasing orders of accuracy. If the solution is constant, the left-hand side of (24) should be zero. This requires the coefficient of $(1/\Delta x)u_i$ to be zero and therefore $\alpha+\beta+\gamma=0$. If the solution is linear, we must have $\alpha-\gamma=1$ to match $u_x|_i$. Finally whilst the first two conditions are necessary for consistency of the approximation in this case we are free to choose the third condition. We can therefore select the coefficient of $(\Delta x/2)\,u_{xx}|_i$ to be zero to improve the accuracy, which means $\alpha+\gamma=0$.

Solving these three equations we find the values $\alpha=1/2$, $\beta=0$ and $\gamma=-(1/2)$ and recover the second-order accurate centred formula

$$u_x|_i=\frac{u_{i+1}-u_{i-1}}{2\Delta x}+O(\Delta x^2).$$

Other approximations can be obtained by selecting a different set of points, for instance, we could have also chosen three points on the side of x_i, e.g., u_i, u_{i-1}, u_{i-2}. The corresponding approximation is known as a one-sided formula. This is sometimes useful to impose boundary conditions on u_x at the ends of the mesh.

3.1.3. Higher-order derivatives

In general, we can derive an approximation of the second derivative using the Taylor expansion

$$\begin{aligned}\frac{\alpha u_{i+1}+\beta u_i+\gamma u_{i-1}}{\Delta x^2} &= (\alpha+\beta+\gamma)\frac{1}{\Delta x^2}u_i+(\alpha-\gamma)\frac{1}{\Delta x}\,u_x|_i \\ &\quad +(\alpha+\gamma)\frac{1}{2}\,u_{xx}|_i+(\alpha-\gamma)\frac{\Delta x}{6}\,u_{xxx}|_i \\ &\quad +(\alpha+\gamma)\frac{\Delta x^2}{12}\,u_{xxxx}|_i+O(\Delta x^4). \qquad (25)\end{aligned}$$

Using similar arguments to those of the previous section we impose

$$\left.\begin{array}{ll} \alpha+\beta+\gamma & =0 \\ \alpha-\gamma & =0 \\ \alpha+\gamma & =2 \end{array}\right\} \Rightarrow \alpha=\gamma=1,\ \beta=-2. \tag{26}$$

The first and second conditions require that there are no u or u_x terms on the right-hand side of Eq. (25) whilst the third conditon ensures that the right-hand side approximates the left-hand side as Δx tens to zero. The solution of Eq. (26) lead us to the second-order centred approximation

$$u_{xx}|_i = \frac{u_{i+1}-2u_i+u_{i-1}}{\Delta x^2} + O(\Delta x^2). \tag{27}$$

The last term in the Taylor expansion $(\alpha-\gamma)\Delta x u_{xxx}|_i/6$ has the same coefficient as the u_x terms and cancels out to make the approximation second-order accurate. This cancellation does not occur if the points in the mesh are not equally spaced. The derivation of a general three point finite difference approximation with unevenly spaced points can also be obtained through Taylor series. We leave this as an exercise for the reader and proceed in the next section to derive a general form using an alternative method.

3.1.4. Finite differences through polynomial interpolation

In this section we seek to approximate the values of $u(x)$ and its derivatives by a polynomial $P(x)$ at a given point x_i. As way of an example we will derive similar expressions to the centred differences presented previously by considering an approximation involving the set of points $\{x_{i-1}, x_i, x_{i+1}\}$ and the corresponding values $\{u_{i-1}, u_i, u_{i+1}\}$. The polynomial of minimum degree that satisfies $P(x_{i-1}) = u_{i-1}$, $P(x_i) = u_i$ and $P(x_{i+1}) = u_{i+1}$ is the quadratic Lagrange polynomial

$$\begin{aligned} P(x) = u_{i-1}&\frac{(x-x_i)(x-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} + u_i\frac{(x-x_{i-1})(x-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})} \\ &+ u_{i+1}\frac{(x-x_{i-1})(x-x_i)}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}. \end{aligned} \tag{28}$$

We can now obtain an approximation of the derivative, $u_x|_i \approx P_x(x_i)$ as

$$\begin{aligned} P_x(x_i) = u_{i-1}&\frac{(x_i-x_{i+1})}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} + u_i\frac{(x_i-x_{i-1})+(x_i-x_{i+1})}{(x_i-x_{i-1})(x_i-x_{i+1})} \\ &+ u_{i+1}\frac{(x_i-x_{i-1})}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)}. \end{aligned} \tag{29}$$

If we take $x_i - x_{i-1} = x_{i+1} - x_i = \Delta x$, we recover the second-order accurate finite difference approximation (14) which is consistent with a quadratic

interpolation. Similarly, for the second derivative we have

$$P_{xx}(x_i) = \frac{2u_{i-1}}{(x_{i-1}-x_i)(x_{i-1}-x_{i+1})} + \frac{2u_i}{(x_i-x_{i-1})(x_i-x_{i+1})} + \frac{2u_{i+1}}{(x_{i+1}-x_{i-1})(x_{i+1}-x_i)} \tag{30}$$

and, again, this approximation leads to the second-order centred finite difference (27) for a constant Δx.

This result is general and the approximation via finite differences can be interpreted as a form of Lagrangian polynomial interpolation. The order of the interpolated polynomial is also the order of accuracy of the finite diference approximation using the same set of points. This is also consistent with the interpretation of a Taylor expansion as an interpolating polynomial.

3.1.5. Finite difference solution of PDEs

We consider the FDM approximation to the solution of the elliptic equation $u_{xx} = s(x)$ in the region $\Omega = \{x : 0 \leq x \leq 1\}$. Discretizing the region using N points with constant mesh spacing $\Delta x = (1/N - 1)$ or $x_i = (i - 1/N - 1)$, we consider two cases with different sets of boundary conditions:

1. $u(0) = \alpha_1$ and $u(1) = \alpha_2$, and
2. $u(0) = \alpha_1$ and $u_x(1) = g$.

In both cases we adopt a centred finite approximation in the interior points of the form

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = s_i; \quad i = 2, \ldots, N-1. \tag{31}$$

The treatment of the first case is straightforward as the boundary conditions are easily specified as $u_1 = \alpha_1$ and $u_N = \alpha_2$. These two conditions together with the $N-2$ equations (31) result in the linear system of N equations with N unknowns represented by

$$\begin{bmatrix} 1 & 0 & & \ldots & & & 0 \\ 1 & -2 & 1 & 0 & \ldots & & 0 \\ 0 & 1 & -2 & 1 & 0 & \ldots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & \ldots & 0 & 1 & -2 & 1 & 0 \\ 0 & & \ldots & 0 & 1 & -2 & 1 \\ 0 & & & \ldots & & 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \\ u_N \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \Delta x^2 s_2 \\ \Delta x^2 s_3 \\ \vdots \\ \Delta x^2 s_{N-2} \\ \Delta x^2 s_{N-1} \\ \alpha_2 \end{bmatrix}.$$

This matrix system can be written in abridged form as $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{s}$. The matrix $\boldsymbol{A}$ is non-singular and admits a unique solution $\boldsymbol{u}$. This is the case for most discretization of well-posed elliptic equations.

In the second case the boundary condition $u(0) = \alpha_1$ is treated in the same way by setting $u_1 = \alpha_1$. The treatment of the Neumann boundary condition $u_x(1) = g$ requires a more careful consideration. One possibility is to use a one-sided approximation of $u_x(1)$ to obtain

$$u_x(1) \approx \frac{u_N - u_{N-1}}{\Delta x} = g. \tag{32}$$

This expression is only first-order accurate and thus inconsistent with the approximation used at the interior points. Given that the PDE is elliptic, this error could potentially reduce the global accuracy of the solution. The alternative is to use a second-order centred approximation

$$u_x(1) \approx \frac{u_{N+1} - u_{N-1}}{\Delta x} = g. \tag{33}$$

Here the value u_{N+1} is not available since it is not part of our discrete set of values but we could use the finite difference approximation at x_N given by

$$\frac{u_{N+1} - 2u_N + u_{N-1}}{\Delta x^2} = s_N$$

and include the Neumann boundary condition (33) to obtain

$$u_N - u_{N-1} = \frac{1}{2}(g\Delta x - s_N \Delta x^2). \tag{34}$$

It is easy to verify that the introduction of either of the Neumann boundary conditions (32) or (34) leads to non-symmetric matrices.

3.2. Time Integration

In this section we address the problem of solving time-dependent PDEs in which the solution is a function of space and time $u(x, t)$. Consider for instance the heat equation

$$u_t - bu_{xx} = s(x) \text{ in } \Omega = \{x, t : 0 \leq x \leq 1, 0 \leq t \leq T\}$$

with an initial condition $u(x, 0) = u_0(x)$ and time-dependent boundary conditions $u(0, t) = \alpha_1(t)$ and $u(1, t) = \alpha_2(t)$, where α_1 and α_2 are known

functions of t. Assume, as before, a mesh or spatial discretization of the domain $\{x_1, \ldots, x_N\}$.

3.2.1. Method of lines

In this technique we assign to our mesh a set of values that are functions of time $u_i(t) = u(x_i, t)$; $i = 1, \ldots, N$. Applying a centred discretization to the spatial derivative of u leads to a system of ordinary differential equations (ODEs) in the variable t given by

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{b}{x^2}\{u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)\} + s_i; \quad i = 2, \ldots, N-1$$

with $u_1 = \alpha_1(t)$ and $u_N = \alpha_2(t)$. This can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} = \frac{b}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} s_2 + \dfrac{b\alpha_1(t)}{\Delta x^2} \\ s_3 \\ \vdots \\ s_{N-2} \\ s_{N-1} + \dfrac{b\alpha_2(t)}{\Delta x^2} \end{bmatrix}$$

or in matrix form as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}(t) = \mathbf{A}\,\mathbf{u}(t) + \mathbf{s}(t). \tag{35}$$

Equation (35) is referred to as the *semi-discrete* form or the method of lines. This system can be solved by any method for the integration of initial-value problems [3]. The numerical stability of time integration schemes depends on the eigenvalues of the matrix $\mathbf{A}$ which results from the space discretization. For this example, the eigenvalues vary between 0 and $-(4\alpha/\Delta x^2)$ and this could make the system very *stiff*, i.e., with large differences in eigenvalues, as $\Delta x \to 0$.

3.2.2. Finite differences in time

The method of finite differences can be applied to time-dependent problems by considering an independent discretization of the solution $u(x, t)$ in space and time. In addition to the spatial discretization $\{x_1, \ldots, x_N\}$, the discretization in time is represented by a sequence of times $t^0 = 0 < \cdots < t^n < \cdots < T$. For simplicity we will assume constant intervals Δx and Δt in space and time, respectively. The discrete solution at a point will be represented by

$u_i^n \approx u(x_i, t^n)$ and the finite difference approximation of the time derivative follows the procedures previously described. For example, the forward difference in time is given by

$$u_t(x, t^n) \approx \frac{u(x, t^{n+1}) - u(x, t^n)}{\Delta t}$$

and the backward difference in time is

$$u_t(x, t^{n+1}) \approx \frac{u(x, t^{n+1}) - u(x, t^n)}{\Delta t}$$

both of which are first-order accurate, i.e., $\epsilon_{\mathrm{T}} = O(\Delta t)$.

Returning to the heat equation $u_t - bu_{xx} = 0$ and using a centred approximation in space, different schemes can be devised depending on the time at which the equation is discretized. For instance, the use of forward differences in time leads to

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{b}{\Delta x^2}\left(u_{i-1}^n - 2u_i^n + u_{i+1}^n\right). \tag{36}$$

This scheme is *explicit* as the values of the solution at time t^{n+1} are obtained directly from the corresponding (known) values at time t^n. If we use backward differences in time, the resulting scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{b}{\Delta x^2}\left(u_{i-1}^{n+1} - 2u_i^{n+1} + u_{i+1}^{n+1}\right). \tag{37}$$

Here to obtain the values at t^{n+1} we must solve a tri-diagonal system of equations. This type of schemes are referred to as *implicit* schemes.

The higher cost of the implicit schemes is compensated by a greater numerical stability with respect to the explicit schemes which are stable in general only for some combinations of Δx and Δt.

3.3. Discretizations Based on the Integral Form

The FDM uses the strong or differential form of the governing equations. In the following, we introduce two alternative methods that use their integral form counterparts: the finite element and the finite volume methods. The use of integral formulations is advantageous as it provides a more natural treatment of Neumann boundary conditions as well as that of discontinuous source terms due to their reduced requirements on the regularity or smoothness of the solution. Moreover, they are better suited than the FDM to deal with complex geometries in multi-dimensional problems as the integral formulations do not rely in any special mesh structure.

These methods use the integral form of the equation as the starting point of the discretization process. For example, if the strong form of the PDE is $\mathcal{L}(u) = s$, the integral from is given by

$$\int_0^1 \mathcal{L}(u) w(x)\,\mathrm{d}x = \int_0^1 s\, w(x)\,\mathrm{d}x \tag{38}$$

where the choice of the weight function $w(x)$ defines the type of scheme.

3.3.1. *The finite element method (FEM)*

Here we discretize the region of interest $\Omega = \{x : 0 \leq x \leq 1\}$ into $N-1$ subdomains or elements $\Omega_i = \{x : x_{i-1} \leq x \leq x_i\}$ and assume that the approximate solution is represented by

$$u^\delta(x,t) = \sum_{i=1}^{N} u_i(t) N_i(x)$$

where the set of functions $N_i(x)$ is known as the expansion basis. Its *support* is defined as the set of points where $N_i(x) \neq 0$. If the support of $N_i(x)$ is the whole interval, the method is called a *spectral method*. In the following we will use expansion bases with compact support which are piecewise continuous polynomials within each element as shown in Fig. 6.

The global shape functions $N_i(x)$ can be split within an element into two local contributions of the form shown in Fig. 7. These individual functions are referred to as the *shape functions* or *trial* functions.

3.3.2. *Galerkin FEM*

In the Galerkin FEM method we set the weight function $w(x)$ in Eq. (38) to be the same as the basis function $N_i(x)$, i.e., $w(x) = N_i(x)$.

Consider again the elliptic equation $\mathcal{L}(u) = u_{xx} = s(x)$ in the region Ω with boundary conditions $u(0) = \alpha$ and $u_x(1) = g$. Equation (38) becomes

$$\int_0^1 w(x) u_{xx}\,\mathrm{d}x = \int_0^1 w(x) s(x)\,\mathrm{d}x.$$

At this stage, it is convenient to integrate the left-hand side by parts to get the weak form

$$-\int_0^1 w_x\, u_x\,\mathrm{d}x + w(1)\, u_x(1) - w(0)\, u_x(0) = \int_0^1 w(x)\, s(x)\,\mathrm{d}x. \tag{39}$$

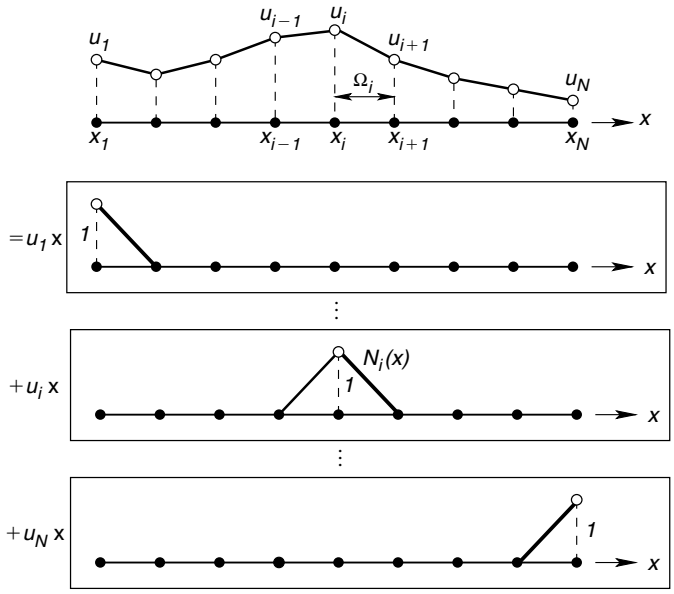


Figure 6. A piecewise linear approximation $u^\delta(x,t) = \sum_{i=1}^{N} u_i(t) N_i(x)$.

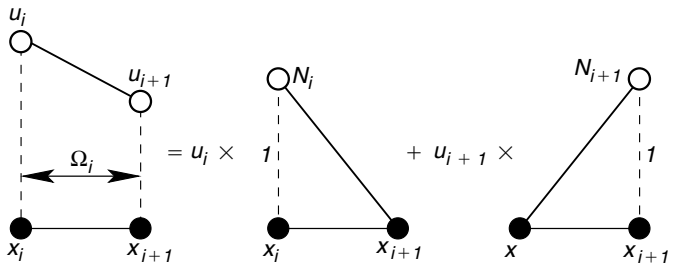


Figure 7. Finite element expansion bases.

This is a common technique in the FEM because it reduces the smoothness requirements on u and it also makes the matrix of the discretized system symmetric. In two and three dimensions we would use Gauss' divergence theorem to obtain a similar result.

The application of the boundary conditions in the FEM deserves attention. The imposition of the Neumann boundary condition $u_x(1) = g$ is straightforward, we simply substitute the value in Eq. (39). This is a very natural way of imposing Neumann boundary conditions which also leads to symmetric

matrices, unlike the FDM. The Dirichlet boundary condition $u(0)=\alpha$ can be applied by imposing $u_1=\alpha$ and requiring that $w(0)=0$. In general, we will impose that the weight functions $w(x)$ are zero at the Dirichlet boundaries.

Letting $u(x)\approx u^\delta(x)=\sum_{j=1}^N u_j N_j(x)$ and $w(x)=N_i(x)$ then Eq. (39) becomes

$$-\int_0^1 \frac{\mathrm{d}N_i}{\mathrm{d}x}(x)\sum_{j=1}^N u_j \frac{\mathrm{d}N_j}{\mathrm{d}x}(x)\,\mathrm{d}x = \int_0^1 N_i(x)\,s(x)\,\mathrm{d}x \tag{40}$$

for $i=2,\ldots,N$. This represents a linear system of $N-1$ equations with $N-1$ unknowns: $\{u_2,\ldots,u_N\}$. Let us proceed to calculate the integral terms corresponding to the ith equation. We calculate the integrals in Eq. (40) as sums of integrals over the elements Ω_i. The basis functions have compact support, as shown in Fig. 6. Their value and their derivatives are different from zero only on the elements containing the node i, i.e.,

$$N_i(x)=\begin{cases}\dfrac{x-x_{i-1}}{\Delta x_{i-1}} & x_{i-1}<x<x_i\\[2ex] \dfrac{x_{i+1}-x}{\Delta x_i} & x_i<x<x_{i+1}\end{cases}$$

$$\frac{\mathrm{d}N_i(x)}{\mathrm{d}x}=\begin{cases}\dfrac{1}{\Delta x_{i-1}} & x_{i-1}<x<x_i\\[2ex] \dfrac{-1}{\Delta x_i} & x_i<x<x_{i+1}\end{cases}$$

with $\Delta x_{i-1}=x_i-x_{i-1}$ and $\Delta x_i=x_{i+1}-x_i$. This means that the only integrals different from zero in (40) are

$$-\int_{x_{i-1}}^{x_i}\frac{\mathrm{d}N_i}{\mathrm{d}x}\left(u_{i-1}\frac{\mathrm{d}N_{i-1}}{\mathrm{d}x}+u_i\frac{\mathrm{d}N_i}{\mathrm{d}x}\right)-\int_{x_i}^{x_{i+1}}\frac{\mathrm{d}N_i}{\mathrm{d}x}\left(u_i\frac{\mathrm{d}N_i}{\mathrm{d}x}+u_{i+1}\frac{\mathrm{d}N_{i+1}}{\mathrm{d}x}\right)\mathrm{d}x$$
$$=\int_{x_{i-1}}^{x_i}N_i s\,\mathrm{d}x+\int_{x_i}^{x_{i+1}}N_i s\,\mathrm{d}x \tag{41}$$

The right-hand side of this equation expressed as

$$F=\int_{x_{i-1}}^{x_i}\frac{x-x_{i-1}}{\Delta x_{i-1}}\,s(x)\,\mathrm{d}x+\int_{x_i}^{x_{i+1}}\frac{x_{i+1}-x}{\Delta x_i}\,s(x)\,\mathrm{d}x$$

can be evaluated using a simple integration rule like the trapezium rule

$$\int_{x_i}^{x_{i+1}} g(x)\,\mathrm{d}x\approx\frac{g(x_i)+g(x_{i+1})}{2}\,\Delta x_i$$

and it becomes

$$F = \left(\frac{\Delta x_{i-1}}{2} + \frac{\Delta x_i}{2} \right) s_i .$$

Performing the required operations in the left-hand side of Eq. (41) and including the calculated valued of F leads to the FEM discrete form of the equation as

$$-\frac{u_i - u_{i-1}}{\Delta x_{i-1}} + \frac{u_{i+1} - u_i}{\Delta x_i} = \frac{\Delta x_{i-1} + \Delta x_i}{2} s_i .$$

Here if we assume that $\Delta x_{i-1} = \Delta x_i = \Delta x$ then the equispaced approximation becomes

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x} = \Delta x \, s_i$$

which is identical to the finite difference formula. We note, however, that the general FE formulation did not require the assumption of an equispaced mesh.

In general the evaluation of the integral terms in this formulation are more efficiently implemented by considering most operations in a standard element $\Omega_{st} = \{-1 \leq x \leq 1\}$ where a mapping is applied from the element Ω_i to the standard element Ω_{st}. For more details on the general formulation see Ref. [4].

3.3.3. Finite volume method (FVM)

The integral form of the one-dimensional linear advection equation is given by Eq. (1) with $f(u) = au$ and $S = 0$. Here the region of integration is taken to be a *control volume* Ω_i, associated with the point of coordinate x_i, represented by $x_{i-(1/2)} \leq x \leq x_{i+(1/2)}$, following the notation of Fig. 4, and the integral form is written as

$$\int_{x_{i-(1/2)}}^{x_{i+(1/2)}} u_t \, \mathrm{d}x + \int_{x_{i-(1/2)}}^{x_{i+(1/2)}} f_x(u) \, \mathrm{d}x = 0. \tag{42}$$

This expression could also been obtained from the weighted residual form (4) by selecting a weight $w(x)$ such that $w(x) = 1$ for $x_{i-(1/2)} \leq x \leq x_{i+(1/2)}$ and $w(x) = 0$ elsewhere. The last term in Eq. (42) can be evaluated analytically to obtain

$$\int_{x_{i-(1/2)}}^{x_{i+(1/2)}} f_x(u) \, \mathrm{d}x = f\left(u_{i+(1/2)}\right) - f\left(u_{i-(1/2)}\right)$$

and if we approximate the first integral using the mid-point rule we finally have the semi-discrete form

$$u_t|_i \left(x_{i+(1/2)} - x_{i-(1/2)}\right) + f\left(u_{i+(1/2)}\right) - f\left(u_{i-(1/2)}\right) = 0.$$

This approach produces a *conservative* scheme if the flux on the boundary of one cell equals the flux on the boundary of the adjacent cell. Conservative schemes are popular for the discretization of hyperbolic equations since, if they converge, they can be proven (Lax-Wendroff theorem) to converge to a weak solution of the conservation law.

3.3.4. Comparison of FVM and FDM

To complete our comparison of the different techniques we consider the FVM discretization of the elliptic equation $u_{xx} = s$. The FVM integral form of this equation over a control volume $\Omega_i = \{x_{i-(1/2)} \leq x \leq x_{i+(1/2)}\}$ is

$$\int_{x_{i-(1/2)}}^{x_{i+(1/2)}} u_{xx}\, \mathrm{d}x = \int_{x_{i-(1/2)}}^{x_{i+(1/2)}} s\, \mathrm{d}x.$$

Evaluating exactly the left-hand side and approximating the right-hand side by the mid-point rule we obtain

$$u_x\left(x_{i+(1/2)}\right) - u_x\left(x_{i-(1/2)}\right) = \left(x_{i+(1/2)} - x_{i-(1/2)}\right)\, s_i. \tag{43}$$

If we approximate $u(x)$ as a linear function between the mesh points $i-1$ and i, we have

$$u_x|_{i-(1/2)} \approx \frac{u_i - u_{i-1}}{x_i - x_{i-1}}, \qquad u_x|_{i+(1/2)} \approx \frac{u_{i+1} - u_i}{x_{i+1} - x_i},$$

and introducing these approximations into Eq. (43) we now have

$$\frac{u_{i+1} - u_i}{x_{i+1} - x_i} - \frac{u_i - u_{i-1}}{x_i - x_{i-1}} = \left(x_{i+(1/2)} - x_{i-(1/2)}\right) s_i.$$

If the mesh is equispaced then this equation reduces to

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x} = \Delta x\, s_i,$$

which is the same as the FDM and FEM on an equispaced mesh.

Once again we see the similarities that exist between these methods although some assumptions in the construction of the FVM have been made. FEM and FVM allow a more general approach to non-equispaced meshes (although this can also be done in the FDM). In two and three dimensions, curvature is more naturally dealt with in the FVM and FEM due to the integral nature of the equations used.

4. High Order Discretizations: Spectral Element/p-Type Finite Elements

All of the approximations methods we have discussed this far have dealt with what is typically known as the h-type approximation. If $h = \Delta x$ denotes the size of a finite difference spacing or finite elemental regions then convergence of the discrete approximation to the PDE is achieved by letting $h \to 0$. An alternative method is to leave the mesh spacing fixed but to increase the polynomial order of the local approximation which is typically denoted by p or the p-type extension.

We have already seen that higher order finite difference approximations can be derived by fitting polynomials through more grid points. The drawback of this approach is that the finite difference stencil gets larger as the order of the polynomial approximation increases. This can lead to difficulties when enforcing boundary conditions particularly in multiple dimensions. An alternative approach to deriving high order finite differences is to use compact finite differences where a Padé approximation is used to approximate the derivatives.

When using the finite element method in an integral formulation, it is possible to develop a compact high-order discretization by applying higher order polynomial expansions within every elemental region. So instead of using just a linear element in each piecewise approximation of Fig. 6 we can use a polynomial of order p. This technique is commonly known as *p-type finite element* in structural mechanics or the *spectral element* method in fluid mechanics. The choice of the polynomial has a strong influence on the numerical conditioning of the approximation and we note that the choice of an equi-spaced Lagrange polynomial is particularly bad for $p > 5$. The two most commonly used polynomial expansions are Lagrange polynomial based on the Gauss–Lobatto–Legendre quadratures points or the integral of the Legendre polynomials in combination with the linear finite element expansion. These two polynomial expansions are shown in Fig. 8. Although this method is more

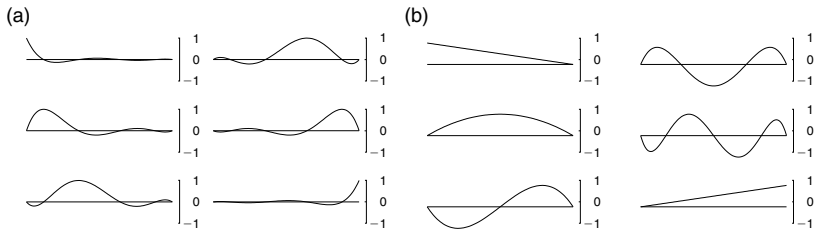


Figure 8. Shape of the fifth order ($p = 5$) polynomial expansions typically used in (a) spectral element and (b) p-type finite element methods.

involved to implement, the advantage is that for a smooth problem (i.e., one where the derivatives of the solution are well behaved) the computational cost increases algebraically whilst the error decreases exponentially fast. Further details on these methods can be found in Refs. [5, 6].

5. Numerical Difficulties

The discretization of linear elliptic equations with either FD, FE or FV methods leads to non-singular systems of equations that can easily solved by standard methods of solution. This is not the case for time-dependent problems where numerical errors may grow unbounded for some discretization. This is perhaps better illustrated with some examples.

Consider the parabolic problem represented by the diffusion equation $u_t - u_{xx} = 0$ with boundary conditions $u(0) = u(1) = 0$ solved using the scheme (36) with $b = 1$ and $\Delta x = 0.1$. The results obtained with $\Delta t = 0.004$ and 0.008 are depicted in Figs. 9(a) and (b), respectively. The numerical solution (b) corresponding to $\Delta t = 0.008$ is clearly unstable.

A similar situation occurs in hyperbolic problems. Consider the one-dimensional linear advection equation $u_t + a u_x = 0$; with $a > 0$ and various explicit approximations, for instance the backward in space, or upwind, scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0 \quad \Rightarrow \quad u_i^{n+1} = (1-\sigma)u_i^n + \sigma u_{i-1}^n, \tag{44}$$

the forward in space, or downwind, scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_i^n}{\Delta x} = 0 \quad \Rightarrow \quad u_i^{n+1} = (1+\sigma)u_i^n - \sigma u_{i+1}^n, \tag{45}$$

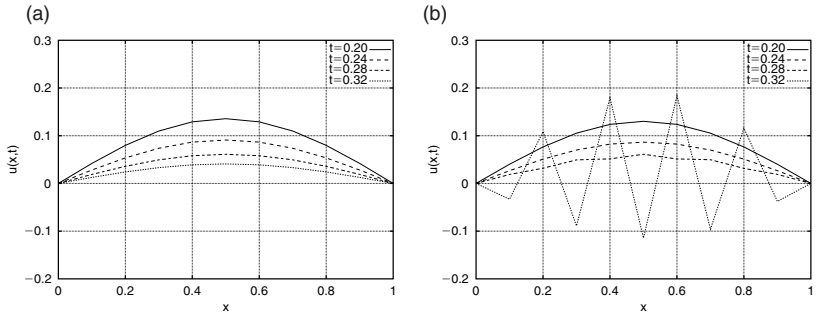


Figure 9. Solution to the diffusion equation $u_t + u_{xx} = 0$ using a forward in time and centred in space finite difference discretization with $\Delta x = 0.1$ and (a) $\Delta t = 0.004$, and (b) $\Delta t = 0.008$. The numerical solution in (b) is clearly unstable.

$$u(x,0) = \begin{cases} 0 & x \le -0.2 \\ 1+5x & -0.2 \le x \le 0 \\ 1-5x & 0 \le x \le 0.2 \\ 0 & x \ge 0.2 \end{cases}$$

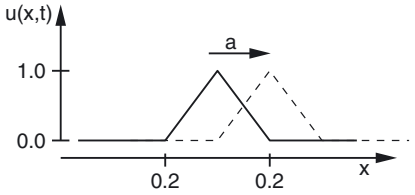


Figure 10. A triangular wave as initial condition for the advection equation.

and, finally, the centred in space is given by

$$\frac{u_i^{n+1}-u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} = 0 \quad \Rightarrow \quad u_i^{n+1} = u_i^n - \frac{\sigma}{2}(u_{i+1}^n - u_{i-1}^n) \tag{46}$$

where $\sigma = (a\Delta t/\Delta x)$ is known as the *Courant number*. We will see later that this number plays an important role in the stability of hyperbolic equations. Let us obtain the solution of $u_t + au_x = 0$ for all these schemes with the initial condition given in Fig. 10.

As also indicated in Fig. 10, the exact solution is the propagation of this wave form to the right at a velocity a. Now we consider the solution of the three schemes at two different Courant numbers given by $\sigma = 0.5$ and 1.5. The results are presented in Fig. 11 and we observe that only the upwinded scheme when $\sigma \le 1$ gives a stable, although diffusive, solution. The centred scheme when $\sigma = 0.5$ appears almost stable but the oscillations grow in time leading to an unstable solution.

6. Analysis of Numerical Schemes

We have seen that different parameters, such as the Courant number, can effect the stability of a numerical scheme. We would now like to set up a more rigorous framework to analyse a numerical scheme and we introduce the concepts of *consistency*, *stability* and *Convergence* of a numerical scheme.

6.1. Consistency

A numerical scheme is consistent if the discrete numerical equation tends to the exact differential equation as the mesh size (represented by Δx and Δt) tends to zero.

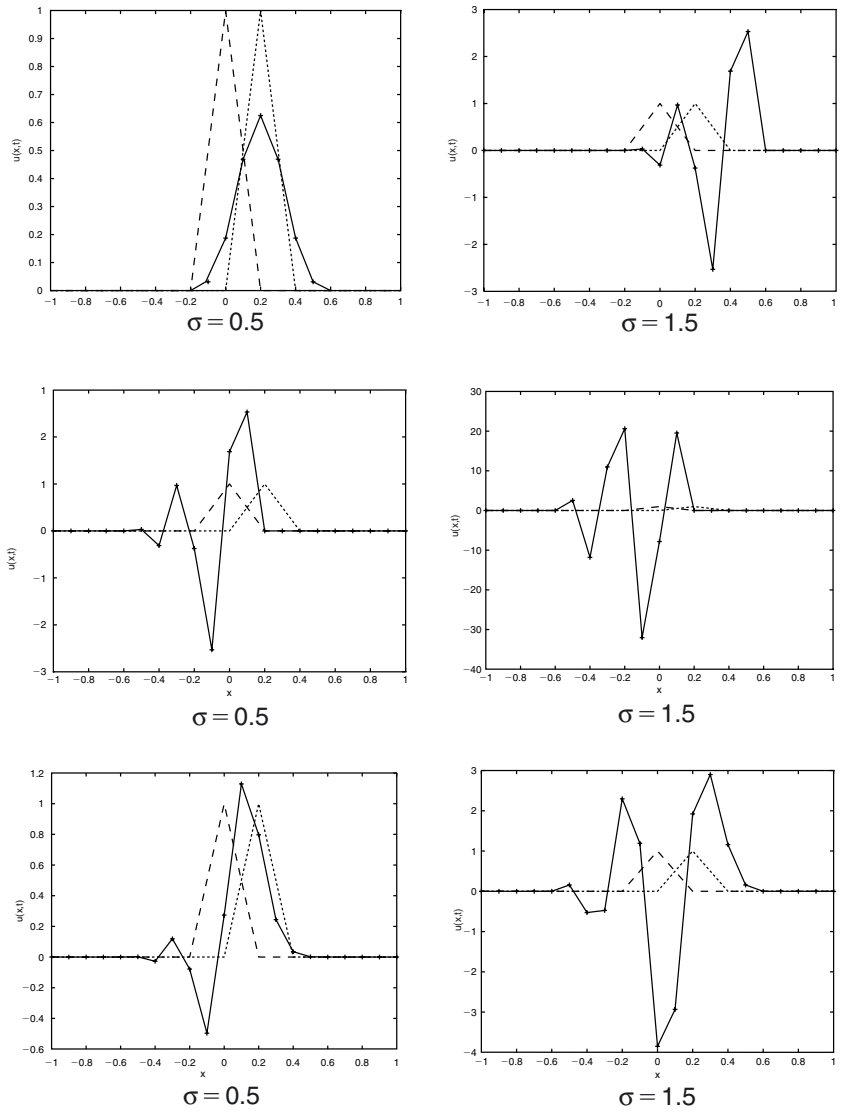


Figure 11. Numerical solution of the advection equation $u_t + au_x = 0$. Dashed lines: initial condition. Dotted lines: exact solution. Solid line: numerical solution.

Consider the centred in space and forward in time finite diference approximation to the linear advection equation $u_t + au_x = 0$ given by Eq. (46). Let us consider Taylor expansions of u_i^{n+1}, u_{i+1}^n and u_{i-1}^n around (x_i, t^n) as

$$u_i^{n+1} = u_i^n + \Delta t\, u_t|_i^n + \frac{\Delta t^2}{2} u_{tt}|_i^n + \cdots$$

$$u_{i+1}^n = u_i^n + \Delta x\, u_x|_i^n + \frac{\Delta x^2}{2} u_{xx}|_i^n + \frac{\Delta x^3}{6} u_{xxx}|_i^n + \cdots$$
$$u_{i-1}^n = u_i^n - \Delta x\, u_x|_i^n + \frac{\Delta x^2}{2} u_{xx}|_i^n - \frac{\Delta x^3}{6} u_{xxx}|_i^n + \cdots$$

Substituting these expansions into Eq. (46) and suitably re-arranging the terms we find that

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x} - (u_t + a u_x)|_i^n = \epsilon_{\mathrm{T}} \tag{47}$$

where ϵ_{T} is known as the *truncation error* of the approximation and is given by

$$\epsilon_{\mathrm{T}} = \frac{\Delta t}{2} u_{tt}|_i^n + \frac{\Delta x^2}{6} a u_{xxx}|_i^n + O(\Delta t^2, \Delta x^4).$$

The left-hand side of this equation will tend to zero as Δt and Δx tend to zero. This means that the numerical scheme (46) tends to the exact equation at point x_i and time level t^n and therefore this approximation is *consistent*.

6.2. Stability

We have seen in the previous numerical examples that errors in numerical solutions can grow uncontrolled and render the solution meaningless. It is therefore sensible to require that the solution is stable, this is that the difference between the computed solution and the exact solution of the discrete equation should remain bounded as $n \to \infty$ for a given Δx.

6.2.1. The Courant–Friedrichs–Lewy (CFL) condition

This is a necessary condition for stability of explicit schemes devised by Courant, Friedrichs and Lewy in 1928.

Recalling the theory of characteristics for hyperbolic systems, the *domain of dependence of a PDE* is the portion of the domain that influences the solution at a given point. For a scalar conservation law, it is the characteristic passing through the point, for instance, the line PQ in Fig. 12. The *domain of dependence of a FD scheme* is the set of points that affect the approximate solution at a given point. For the upwind scheme, the numerical domain of dependence is shown as a shaded region in Fig. 12.

The *CFL criterion* states that a *necessary* condition for an explicit FD scheme to solve a hyperbolic PDE to be stable is that, for each mesh point, the domain of dependence of the FD approximation contains the domain of dependence of the PDE.

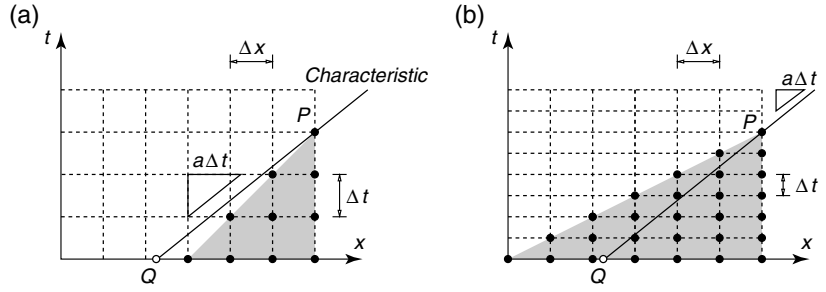


Figure 12. Solution of the advection equation by the upwind scheme. Physical and numerical domains of dependence: (a) $\sigma = (a\Delta t/\Delta x) > 1$, (b) $\sigma \leq 1$.

For a Courant number $\sigma = (a\Delta t/\Delta x)$ greater than 1, changes at Q will affect values at P but the FD approximation cannot account for this.

The CFL condition is necessary for stability of explicit schemes but *it is not sufficient*. For instance, in the previous schemes we have that the upwind FD scheme is stable if the CFL condition $\sigma \leq 1$ is imposed. The downwind FD scheme does not satisfy the CFL condition and is unstable. However, the centred FD scheme is unstable even if $\sigma \leq 1$.

6.2.2. *Von Neumann (or Fourier) analysis of stability*

The stability of FD schemes for hyperbolic and parabolic PDEs can be analysed by the von Neumann or Fourier method. The idea behind the method is the following. As discussed previously the analytical solutions of the model diffusion equation $u_t - b\,u_{xx} = 0$ can be found in the form

$$u(x,t) = \sum_{m=-\infty}^{\infty} \mathrm{e}^{\beta_m t}\,\mathrm{e}^{I k_m x}$$

if $\beta_m + b\,k_m^2 = 0$. This solution involves a Fourier series in space and an exponential decay in time since $\beta_m \leq 0$ for $b > 0$. Here we have included the complex version of the Fourier series, $\mathrm{e}^{I k_m x} = \cos\, k_m x + I\, \sin\, k_m x$ with $I = \sqrt{-1}$, because this simplifies considerably later algebraic manipulations.

To analyze the growth of different Fourier modes as they evolve under the numerical scheme we can consider each frequency separately, namely

$$u(x,t) = \mathrm{e}^{\beta_m t}\,\mathrm{e}^{I k_m x}.$$

A discrete version of this equation is $u_i^n = u(x_i, t^n) = \mathrm{e}^{\beta_m t^n}\mathrm{e}^{Ik_m x_i}$. We can take, without loss of generality, $x_i = i\,\Delta x$ and $t^n = n\,\Delta t$ to obtain

$$u_i^n = \mathrm{e}^{\beta_m n\,\Delta t}\,\mathrm{e}^{Ik_m i\,\Delta x} = \left(\mathrm{e}^{\beta_m \Delta t}\right)^n \mathrm{e}^{Ik_m i\,\Delta x}.$$

The term $\mathrm{e}^{Ik_m i\,\Delta x} = \cos(k_m i\,\Delta x) + I\,\sin(k_m i\,\Delta x)$ is bounded and, therefore, any growth in the numerical solution will arise from the term $G = \mathrm{e}^{\beta_m \Delta t}$, known as the *amplification factor*. Therefore the numerical method will be stable, or the numerical solution u_i^n bounded as $n \to \infty$, if $|G| \leq 1$ for solutions of the form

$$u_i^n = G^n\,\mathrm{e}^{Ik_m i\,\Delta x}.$$

We will now proceed to analyse, using the von Neummann method, the stability of some of the schemes discussed in the previous sections.

Example 1 Consider the explicit scheme (36) for the diffusion equation $u_t - bu_{xx} = 0$ expressed here as

$$u_i^{n+1} = \lambda u_{i-1}^n + (1 - 2\lambda)u_i^n + \lambda u_{i+1}^n; \quad \lambda = \frac{b\,\Delta t}{\Delta x^2}.$$

We assume $u_i^n = G^n\mathrm{e}^{Ik_m i\,\Delta x}$ and substitute in the equation to get

$$G = 1 + 2\lambda\,[\cos(k_m\,\Delta x) - 1]\,.$$

Stability requires $|G| \leq 1$. Using $-2 \leq \cos(k_m\,\Delta x) - 1 \leq 0$ we get $1 - 4\lambda \leq G \leq 1$ and to satisfy the left inequality we impose

$$-1 \leq 1 - 4\lambda \leq G \quad \Longrightarrow \quad \lambda \leq \frac{1}{2}.$$

This means that for a given grid size Δx the maximum allowable timestep is $\Delta t = (\Delta x^2/2b)$.

Example 2 Consider the implicit scheme (37) for the diffusion equation $u_t - bu_{xx} = 0$ expressed here as

$$\lambda u_{i-1}^{n+1} + -(1 + 2\lambda)u_i^{n+1} + \lambda u_{i+1}^{n+1} = -u_i^n\,; \quad \lambda = \frac{b\,\Delta t}{\Delta x^2}.$$

The amplification factor is now

$$G = \frac{1}{1 + \lambda(2 - \cos\,\beta_m)}$$

and we have $|G| < 1$ for any β_m if $\lambda > 0$. This scheme is therefore unconditionally stable for any Δx and Δt. This is obtained at the expense of solving a linear system of equations. However, there will still be restrictions on Δx

and Δt based on the accuracy of the solution. The choice between an explicit or an implicit method is not always obvious and should be done based on the computer cost for achieving the required accuracy in a given problem.

Example 3 Consider the upwind scheme for the linear advection equation $u_t + a u_x = 0$ with $a > 0$ given by

$$u_i^{n+1} = (1-\sigma)u_i^n + \sigma u_{i-1}^n; \quad \sigma = \frac{a\Delta t}{\Delta x}.$$

Let us denote $\beta_m = k_m \Delta x$ and introduce the discrete Fourier expression in the upwind scheme to obtain

$$G = (1-\sigma) + \sigma\, \mathrm{e}^{-I\beta_m}$$

The stability condition requires $|G| \leq 1$. Recall that G is a *complex* number $G = \xi + I\eta$ so

$$\xi = 1 - \sigma + \sigma\, \cos\, \beta_m; \quad \eta = -\sigma\, \sin\, \beta_m$$

This represents a circle of radius σ centred at $1-\sigma$. The stability condition requires the locus of the points (ξ, η) to be interior to a unit circle $\xi^2 + \eta^2 \leq 1$. If $\sigma < 0$ the origin is outside the unit circle, $1 - \sigma > 1$, and the scheme is unstable. If $\sigma > 1$ the back of the locus is outside the unit circle $1 - 2\sigma < 1$ and it is also unstable. Therefore, for stability we require $0 \leq \sigma \leq 1$, see Fig. 13.

Example 4 The forward in time, centred in space scheme for the advection equation is given by

$$u_i^{n+1} = u_i^n - \frac{\sigma}{2}(u_{i+1}^n - u_{i-1}^n); \quad \sigma = \frac{a\Delta t}{\Delta x}.$$

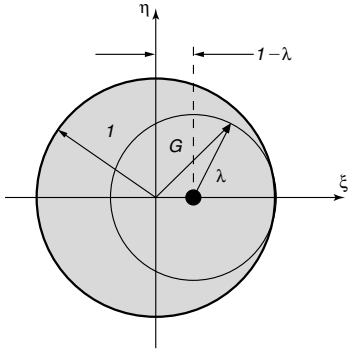


Figure 13. Stability region of the upwind scheme.

The introduction of the discrete Fourier solution leads to

$$G = 1 - \frac{\sigma}{2}(e^{I\beta_m} - e^{-I\beta_m}) = 1 - I\sigma \ \sin \ \beta_m$$

Here we have $|G|^2 = 1 + \sigma^2 \ \sin^2 \ \beta_m > 1$ always for $\sigma \neq 0$ and it is therefore unstable. We will require a different time integration scheme to make it stable.

6.3. Convergence: Lax Equivalence Theorem

A scheme is said to be *convergent* if the difference between the computed solution and the exact solution of the PDE, i.e., the error $E_i^n = u_i^n - u(x_i, t^n)$, vanishes as the mesh size is decreased. This is written as

$$\lim_{\Delta x, \Delta t \to 0} |E_i^n| = 0$$

for fixed values of x_i and t^n. This is the fundamental property to be sought from a numerical scheme but it is difficult to verify directly. On the other hand, consistency and stability are easily checked as shown in the previous sections.

The main result that permits the assessment of the convergence of a scheme from the requirements of consistency and stability is the equivalence theorem of Lax stated here without proof:

Stability is the necessary and sufficient condition for a *consistent* linear FD approximation to a well-posed linear initial-value problem to be *convergent*.

7. Suggestions for Further Reading

The basics of the FDM are presented a very accessible form in Ref. [7]. More modern references are Refs. [8, 9].

An elementary introduction to the FVM can be consulted in the book by Versteeg and Malalasekera [10]. An in-depth treatment of the topic with an emphasis on hyperbolic problems can be found in the book by Leveque [2].

Two well established general references for the FEM are the books of Hughes [4] and Zienkiewicz and Taylor [11]. A presentation from the point of view of structural analysis can be consulted in Cook *et al*. [11]

The application of p-type finite element for structural mechanics is dealt with in book of Szabo and Babuška [5]. The treatment of both p-type and spectral element methods in fluid mechanics can be found in book by Karniadakis and Sherwin [6].

A comprehensive reference covering both FDM, FVM and FEM for fluid dynamics is the book by Hirsch [13]. These topics are also presented using a more mathematical perspective in the classical book by Quarteroni and Valli [14].

References

[1] J. Bonet and R. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 1997.

[2] R. Leveque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002.

[3] W. Cheney and D. Kincaid, *Numerical Mathematics and Computing*, 4th edn., Brooks/Cole Publishing Co., 1999.

[4] T. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publishers, 2000.

[5] B. Szabo and I. Babuška, *Finite Element Analysis*, Wiley, 1991.

[6] G.E. Karniadakis and S. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.

[7] G. Smith, *Numerical Solution of Partial Differential Equations: Finite Diference Methods*, Oxford University Press, 1985.

[8] K. Morton and D. Mayers, *Numerical Solution of Partial Differential Equations*, Cambridge University Press, 1994.

[9] J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Springer-Verlag, 1995.

[10] H. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics. The Finite Volume Method*, Longman Scientific & Technical, 1995.

[11] O. Zienkiewicz and R. Taylor, *The Finite Element Method: The Basis*, vol. 1, Butterworth and Heinemann, 2000.

[12] R. Cook, D. Malkus, and M. Plesha, *Concepts and Applications of Finite Element Analysis*, Wiley, 2001.

[13] C. Hirsch, *Numerical Computation of Internal and External Flows*, vol. 1, Wiley, 1988.

[14] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer-Verlag, 1994.

Chapter 4

Numerical Methods for Linear Conservation Laws

We study the basic theory of numerical methods for solving the linear advection equation and linear hyperbolic systems. The emphasis will be on three important concepts in numerical study, consistency, stability and convergence. We study the basic theory on these focusing on linear equations in this chapter. The fundamental ideas we learn in this chapter will be extended to the nonlinear case.

1. Discretization

We continue to consider our simple model equation in 1D, the linear constant coefficient advection equation,

$$\begin{cases} u_t + au_x = 0, x \in \mathbb{R}, t \geq 0, \\ u(x,0) = u_0(x), \end{cases} \tag{4.1}$$

where $a > 0$ is constant and the advective flux $f(u) = au$. In this chapter, we will frequently (and conveniently) call Eq. 4.1 *the PDE*.

As before, we follow the cell-centered (rather than cell interface-centered) notation for discrete cells x_i and the conventional temporal discretization t^n:

$$x_i = (i - \frac{1}{2})\Delta x, i = 1, ..., N, \tag{4.2}$$

$$t^n = n\Delta t, n = 0, ...M. \tag{4.3}$$

Then the cell interface-centered grid points are written using the 'half-integer' indices:

$$x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}. \tag{4.4}$$

Definition: Let $u_i^n = u(x_i, t^n)$ be the pointwise values of the exact solution of Eq. 4.1 at the discrete points (x_i, t^n). This is the analytical solution of the PDE and satisfies it without any form of numerical errors.

Definition: Let U_i^n be the numerical approximations to the exact solution of the PDE. For instance, U_i^n represents

$$U_i^n = \begin{cases} u_i^n \text{ for FDM, or} \\ \\ \frac{1}{\Delta x}\int_{x_{1-1/2}}^{x_{1+1/2}} u(x,t^n)dx \text{ for FVM} \end{cases} \tag{4.5}$$

Definition: Let D_i^n be the exact solution of the associated 'difference equation (DE)' of the PDE, e.g., the forward in time backward in space (FTBS):

$$\frac{D_i^{n+1} - D_i^n}{\Delta t} = -a\frac{D_i^n - D_{i-1}^n}{\Delta x}. \tag{4.6}$$

Since D_i^n is the exact solution of the DE, there is *no* round-off errors involved. When we study numerical solution of PDEs, the solutions are affected by numerical errors. They mainly come from two sources of numerical errors, and we are now ready to define them.

Definition: The *discretization error* E_d^n at (x_i, t^n) is defined by

$$E_{d,i}^n = u_i^n - D_i^n. \tag{4.7}$$

Definition: The *round-off error* $E_{r,i}^n$ at (x_i, t^n) is defined by

$$E_{r,i}^n = D_i^n - U_i^n. \tag{4.8}$$

Definition: The *global error* $E_{g,i}^n$ at (x_i, t^n) is defined by

$$E_{g,i}^n = u_i^n - U_i^n. \tag{4.9}$$

Note by definition, $E_{g,i}^n = E_{d,i}^n + E_{r,i}^n$.

Definition: We say that the numerical method is *convergent* at t^n in a given norm $||\cdot||$ if

$$\lim_{\Delta x, \Delta t \to 0} ||E_g^n|| = 0. \tag{4.10}$$

Remark: We note that the discretization error $E_{d,i}^n$ is the sum of the truncation error $E_{T,i}^n$ for the DE Eq. 4.6 and any numerical errors $E_{B,i}^n$ introduced by the numerical handling of boundary conditions.

Remark: We define the round-off error $E_{r,i}^n$ by the numerical errors introduced after a repetitive number of arithmetic computer operations in which the computer constantly rounds off the numbers to some significant digits.

Note: In Eq. 4.10, the error norm $||E_g^n||$ is to be understood as a value obtained by taking a norm of the discrete cell-based data, $E_{g,i}^n$ with a choice of norm. It is worth discussing a proper choice of norm when measuring convergence rates in norms for given (discrete) data. We will discuss this matter in more detail in the next section.

2. Choice of Norms

In order to quantify the errors we defined above, we must choose a norm that is most appropriate to the inherent properties of the data we wish to measure. Extended from the L^p-norm in the space of functions, we note that the discrete l^p-norm is a norm in the space of sequences.

One of the properties of the norm space is that all norms are equivalent – the norm equivalence theorem – in finite dimensional vector space (not so in infinite dimensional space though), and therefore we might wonder why making a different choice of norms would make any different outcomes in measuring errors in CFD, since the real applications in CFD are all finite vector space indeed.

The truth is that, if we recall the definition of 'the equivalence of any two norms', it simply means that two norms are bounded with each others using some real constants, i.e., $||\cdot||_\alpha$ and $||\cdot||_\beta$ are equivalent if there exist positive real numbers C and D such that

$$C||x||_\alpha \leq ||x||_\beta \leq D||x||_\alpha. \tag{4.11}$$

Therefore, we should keep in mind that the *equivalence* relation does not mean that the measured quantities (e.g., convergence rate) in different norms are the same! They are simply bounded by each other with some scaling factors. Hence, the norm equivalence can be useful to prove some boundedness property of the numerical solutions, but not so much to quantify them. See Fig. 1 and Fig. 2.

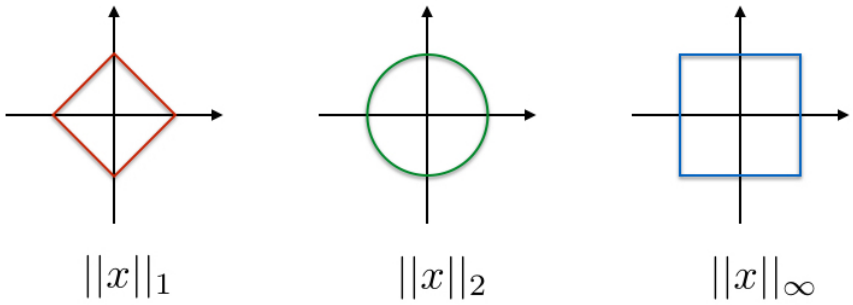
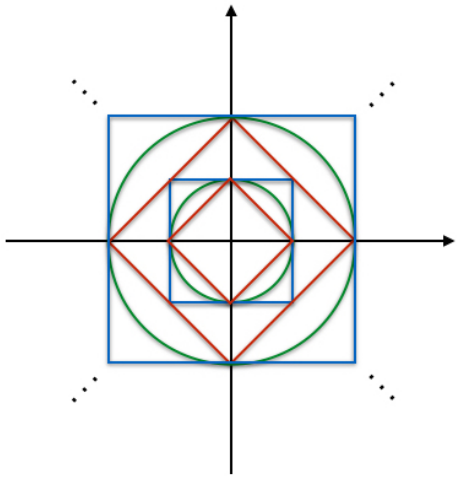


Figure 1. Illustrations of unit circle, $||x|| = 1$, in three different norms: 1-norm, 2-norm and ∞-norm.

Once again, let us define the discrete l^p-norms that are frequently used in CFD.

Definition: The p-norm (or l^p-norm) of the N number of discrete data points E_i^n at $t = t^n$ is defined by

$$||E^n||_p = \left(\Delta x \sum_i^N \left|E_i^n\right|^p\right)^{\frac{1}{p}}, \tag{4.12}$$



$$||x||_1 \leq C_1||x||_2 \leq C_2||x||_\infty \leq C_3||x||_1 \leq C_4||x||_2 \leq C_5||x||_\infty \leq \cdots$$

Figure 2. The norm equivalence theorem indicates any given norm in finite dimensional vector space can be scaled to be bounded in a different choice of norms.

and the max-norm (or l^∞-, or ∞-norm) of the N discrete data points E_i^n at $t = t^n$ is given by

$$||E^n||_\infty = \max_{1 \leq i \leq N} \left| E_i^n \right|. \tag{4.13}$$

Let us now consider using the max-norm when measuring the errors to show the numerical convergence as in Eq. 4.10. By definition, the max-norm represents the maximum value of the 'pointwise' errors. The pointwise errors should behave almost in a similar way at all discrete data points x_i if the errors evaluate smoothly varying data set, e.g., discrete data of sinusoidal waves. In this case, the pointwise error estimation using max-norm is perfectly satisfactory.

However, if the data under consideration involve some form of jumps in discontinuity, such as shock or contact discontinuity, the use of max-norm won't be any good at all and cannot provide any realistic measurement. This is because the error is the largest at at any jumps in discontinuity, and in fact, *all* numerical schemes become only first-order accurate at discontinuities, experiencing significant drops in numerical accuracy. At discontinuities, the numerical solution fails to converge on grid resolutions and there is nothing we can do to improve solution convergence and accuracy. This means that the max-norm will pick up the maximum error at such discontinuity, even though the numerical solutions from the rest of smooth parts are convergent with numerical errors that are usually orders of magnitude smaller than at the discontinuity.

In practice, l^1-norm is very popular for conservation laws (i.e., PDEs in weak form) for hyperbolic PDEs, since by definition, it describes the *integral* quantities of the solutions.

On the other hand, l^2-norm is often used for linear problems because of the utility of Fourier analysis (e.g., Parseval's identity, classical von Neumann stability analysis) in that the Fourier transform $\hat{u}(\xi)$ of $u(x)$ has the same l^2-norm as $u(x)$, $||\hat{u}^n||_2 = ||u^n||_2$. This is useful as it suffices to show that $||\hat{u}^n||_2$ is bounded when showing $||u^n||_2$ is bounded (and vice versa) when we opt to show the growth rate of the solution for an arbitrary wave number ξ is bounded in the Fourier space.

Differences are typically greatest between the max-norm and other choices of norms, and in many practical applications of CFD, the use of l^1- and l^2-norms will give similar results. If this is the case, the choice of norm may just depend mostly on which yields an easier mathematical analysis, e.g., the l^1-norm for (linear and nonlinear) conservation laws, while the l^2-norm for linear equations.

3. The Fundamental Theorem of Numerical Methods – The Lax Equivalence Theorem for Linear PDEs

The ultimate goal in this chapter is to show (at least partially) one of the theorems that is very powerful to provide us great levels of insights in numerical differential equations. Briefly speaking, the theorem says, for linear PDEs,

$$\texttt{consistency + stability} \iff \texttt{convergence}$$

Let us take a moment to think about the meaning of this theorem. It says that if the numerical scheme *converges* to a (weak) solution provided the scheme is proven to be consistent (we are going to define it shortly) and stable. So, what is good about it? The good news is that in numerically solving many PDE systems, it is often very difficult to directly show convergence of a given numerical method because not many PDEs have their exact analytical solutions available (see the definition of convergence in Eq. 4.10). Without guaranteeing the existence of such analytical solutions, one cannot possibly say her/his numerical scheme converges to a mathematically meaningful and correct solution at all.

A nice workaround is instead to look at numerical stability and consistency that are based on a recurrence property of the numerical method acting on the discrete grid data. The Lax Equivalence theorem then indicates that such numerical method is indeed a convergent method that produces a well-defined weak solution. Now let's take a look at this nice theorem in more details.

First, we define few more things.

Definition: Let $\mathcal{N}$ be the (linear) numerical operator mapping the approximate solution at one time step to the approximate solution at the next time step. Then a general explicit numerical method can be written as

$$U_i^{n+1} = \mathcal{N}(U_i^n). \tag{4.14}$$

We define the *one-step error* $E^n_{1step,i}$ by

$$E^n_{1step,i} = u^n_i - \mathcal{N}(u^{n-1}_i), \tag{4.15}$$

and the *local truncation error* $E^n_{LT,i}$ by

$$E^n_{LT,i} = \frac{1}{\Delta t} E^n_{1step,i}. \tag{4.16}$$

We have already discussed the *the order of method* previously, and we now can define it again using the local truncation error.

Definition: We say that the numerical method is *of order p (or pth order accurate)* if for all sufficiently smooth data with compact support, the local truncation error is given as

$$E^n_{LT,i} = \mathcal{O}(\Delta t^p, \Delta x^p). \tag{4.17}$$

Remark: One can obviously introduce a method that has different orders of accuracy in space and time, i.e., a method that is of p-th order accurate in time and r-th order accurate in space can be defined as

$$E^n_{LT,i} = \mathcal{O}(\Delta t^p, \Delta x^r). \tag{4.18}$$

In this case, the numerical solution in a fully resolved state – both temporally and spatially – will exhibit its convergence rate dominated by the lower one between the two, i.e.,

$$E^n_{LT,i} = \mathcal{O}(\Delta t^s) = \min\Big[\mathcal{O}(\Delta t^p), \mathcal{O}(\Delta x^r)\Big]. \tag{4.19}$$

Example: Consider the method of lines. Combining a first-order temporal discretization and a second-order spatial discretization results in a first-order accurate method unless $\Delta t \approx \mathcal{O}(\Delta x^2)$.

Remark: When solving numerical PDEs, it is common to have a numerical method that combines two different orders of accuracy in temporal and spatial discretizations, yielding Eq. 4.18. In many cases, one often has $p \leq r$, and therefore the lower temporal accuracy dominates the overall convergence rate. In this case, the little trick in the previous example can be used to provide a better balance between the spatial and temporal orders, keeping the overall solution to be r-th order accurate (rather than p-th order!) by adopting the following principle:

$$\Delta t \approx \mathcal{O}(\Delta x^{r/p}). \tag{4.20}$$

More generally, while satisfying the numerical stability condition of Δt on a given grid resolution N (i.e., the CFL condition), one can conduct a grid resolution study by changing the size of grid resolutions from N_1 to N_2 (e.g., $N_1 < N_2$) by following the simple rule:

the p-th rate of change in Δt
= the r-th rate of change in Δx,

or mathematically writing,

$$\left(\frac{\Delta t_{N_1}}{\Delta t_{N_2}}\right)^p = \left(\frac{\Delta x_{N_1}}{\Delta x_{N_2}}\right)^r = \left(\frac{N_2}{N_1}\right)^r \tag{4.21}$$

3.1. Consistency

Let's now formally define consistency of the numerical methods.

Definition: We say the numerical method is *consistent* in $||\cdot||$ with a proposed DE if

$$\lim_{\Delta t, \Delta x \to 0} ||E_{LT}^n|| = 0 \tag{4.22}$$

for all smooth functions $u(x,t)$ that satisfies the given PDE.

Remark: In words, the numerical consistency is a measure to see if the numerical operator $\mathcal{N}$ is in fact 'consistent' with the DE of interest in a sense that the method should introduce a small error in any one step.

Remark: On the other hand, the numerical stability is a property that the numerical method does not produce any local errors that grow catastrophically and hence a bound on the global error can be obtained in terms of these local errors.

Example: We consider the first-order upwind method – the DE – for the linear scalar advection – the PDE – and see if the DE is consistent. The upwind DE for $a > 0$ is

$$U_i^{n+1} = U_i^n - a\frac{\Delta t}{\Delta x}\Big(U_i^n - U_{i-1}^n\Big). \tag{4.23}$$

Let's now apply Taylor expansion to obtain the local truncation error $E_{LT,i}^{n+1}$:

$$E_{LT,i}^{n+1} = \frac{1}{\Delta t}\left[u(x_i,t^{n+1}) - \Big\{u(x_i,t^n) - a\frac{\Delta t}{\Delta x}\Big[u(x_i,t^n) - u(x_{i-1},t^n)\Big]\Big\}\right] \tag{4.24}$$

where

$$\mathcal{N}(u_i^n) = u(x_i,t^n) - a\frac{\Delta t}{\Delta x}\Big[u(x_i,t^n) - u(x_{i-1},t^n)\Big] \tag{4.25}$$

Here, please note that we differentiate between u_i^n and U_i^n. Using Taylor expansions of $u(x_i,t^{n+1})$ and $u(x_{i-1},t^n)$:

$$u(x_i,t^{n+1}) = u(x_i,t^n) + u_t(x_i,t^n)\Delta t + u_{tt}(x_i,t^n)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3), \tag{4.26}$$

and

$$u(x_{i-1},t^n) = u(x_i,t^n) - u_x(x_i,t^n)\Delta x + u_{xx}(x_i,t^n)\frac{\Delta x^2}{2} + \mathcal{O}(\Delta x^3). \tag{4.27}$$

Substituting Eq. 4.26 and Eq.4.27 into Eq.4.24 gives

$$E_{LT,i}^{n+1} = u_t(x_i, t^n) + au_x(x_i, t^n) + u_{tt}(x_i, t^n)\frac{\Delta t}{2} - au_{xx}(x_i, t^n)\frac{\Delta x}{2} + \mathcal{O}(\Delta t^2, \Delta x^2). \tag{4.28}$$

Note that the first two terms vanishes since $u(x,t)$ is the exact solution to the PDE. Using so called the Cauchy-Kowalewski procedure, we get

$$u_{tt} = -au_{tx} = -a(-au_x)_x = a^2 u_{xx}, \tag{4.29}$$

and we finally arrive to get

$$E_{LT,i}^{n+1} = \frac{1}{2}a(a-1)u_{xx}(x_i, t^n)\mathcal{O}(\Delta t, \Delta x) + \mathcal{O}(\Delta t^2, \Delta x^2). \tag{4.30}$$

This means that the local truncation error $E_{LT,i}^n$ is dominated by $\mathcal{O}(\Delta t, \Delta x)$, whereby we show that the method is first-order accurate in both space and time. It also proves that the method is consistent because $E_{LT,i}^{n+1}$ approaches to zero when Δt and Δx go to zero as long as $u(x,t)$ is at least twice differentiable in both space and time.

Note: In the above, we actually have the 'pointwise' property of the limit:

$$\lim_{\Delta t, \Delta x \to 0} E_{LT,i}^{n+1} = 0, \tag{4.31}$$

as long as the solution $u(x,t)$ is twice differentiable in space and time. Henceforth, it is natural to see its norm $||E_{LT}^n||$ approaches to zero in the limit, without depending on the choice of norms, even in the max-norm. This is what is to be expected for smooth continous solutions.

Problem 1 The Lax-Friedrichs (LF) method for the our model PDE reads as

$$U_i^{n+1} = \frac{1}{2}\Big(U_{i+1}^n + U_{i-1}^n\Big) - \frac{\Delta t}{2\Delta x}\Big(f(U_{i+1}^n) - f(U_{i-1}^n)\Big). \tag{4.32}$$

where the flux given by $f(u) = au$. As before, assume $a > 0$.
(a) Show that the LF method is consistent.
(b) Rewrite the LF method in the conservative form,

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\Big(\hat{f}(U_{i+1/2}^n) - \hat{f}(U_{i-1/2}^n)\Big). \tag{4.33}$$

That is to say, please find expressions for $\hat{f}_{i\pm 1/2}^n$ as functions of U_k^n and the original flux $f(U_k^n)$, $k = -1, 0, 1$.
(c) Numerically solve the sinusoidal advection problem we had in Chapter 2 using the LF method with the grid resolution $N = 32, 64, 128$ with $C_a = 0.8$. Please show your plots at $t = t_{cycle1}$ and t_{cycle2} at all three grid resolutions. Describe your findings and compare the LF results with the previously studied stable solution using the upwind method.

Problem 2 The Lax-Wendroff (LW) method for the our model PDE reads as

$$U_i^{n+1} = U_i^n - \frac{C_a}{2}\Big(U_{i+1}^n - U_{i-1}^n\Big) + \frac{C_a^2}{2}\Big(U_{i+1}^n - 2U_i^n + U_{i-1}^n\Big). \tag{4.34}$$

Again, let us assume $a > 0$.
(a) Use the method called Cauchy-Kowalewski (or the Lax-Wendroff technique) to derive the LW method, that is,
(a)-(i) First show using Taylor expansion and using the conversion between the spatial and temporal derivatives

$$u_t = -f_x = -f_u u_x = -a u_x, \text{etc.}, \tag{4.35}$$

$$u(x_i, t^{n+1}) = u(x_i, t^n) - \Delta t f_x + \frac{\Delta t^2}{2}(a f_x)_x + \mathcal{O}(\Delta t^3). \tag{4.36}$$

(a)-(ii) Then finally use the second-order central differencing for the spatial derivatives to obtain the relation in Eq. 4.34.
(b) Show that the LW method is consistent.
(c) Rewrite the LW method in the conservative form by introducing the conservative flux $\hat{f}_{i\pm1/2}$ as in Eq. 4.33.
(d) Numerically solve the sinusoidal advection problem we had in Chapter 2 using the LW method with the grid resolution $N = 32, 64, 128$ with $C_a = 0.8$. Please show your plots at $t = t_{cycle1}$ and t_{cycle2} at all three grid resolutions. Describe your findings and compare the LW results with the previously studied stable solutions using the upwind method and the LF method.

3.2. Stability Theory

The form of stability bounds in this section provides a useful information in analyzing 'linear' methods. It has to be emphasized that for 'nonlinear' methods, the same technique we adopt for the linear method becomes hard to apply, and therefore one has to provide a different approach to discuss nonlinear stability (we will study such approach(es) later!). We limit our interest in the linear stability theory in this chapter.

In order to assess stability of the linear PDEs, we essentially need to bound the global error $E_{g,i}^n = u_i^n - U_i^n$ using a recurrence relation. Applying the *linear* numerical operator $\mathcal{N}$ to U_i^n, we obtain

$$U_i^{n+1} = \mathcal{N}(U_i^n) = \mathcal{N}(u_i^n - E_{g,i}^n). \tag{4.37}$$

The global error at t^{n+1} is now

$$\begin{aligned} E_{g,i}^{n+1} &= u_i^{n+1} - U_i^{n+1} && (4.38)\\ &= u_i^{n+1} - \mathcal{N}(u_i^n - E_{g,i}^n) && (4.39)\\ &= u_i^{n+1} - \mathcal{N}(u_i^n) - \Big(\mathcal{N}(u_i^n - E_{g,i}^n) - \mathcal{N}(u_i^n)\Big) && (4.40)\\ &= \Delta t E_{LT,i}^{n+1} - \Big(\mathcal{N}(u_i^n - E_{g,i}^n) - \mathcal{N}(u_i^n)\Big). && (4.41) \end{aligned}$$

Note that the first term in Eq. 4.41 is the new one-step error introduced in this time step, and this term is therefore related to the consistency control of the numerical method. On the other hand, the second term in the parenthesis is the effect of the numerical method on the *previous* global error $E^n_{g,i}$ and this is the term that is to do with the stability control.

Definition: We say the linear numerical method defined by the linear operator $\mathcal{N}$ is *stable* in $||\cdot||$ if there is a constant C such that

$$||\mathcal{N}^n|| \leq C, \forall n\Delta t \leq T, \tag{4.42}$$

for each time T.

Note: We note here that the superscript n on $\mathcal{N}$ represents *powers* of the matrix (or linear operator) obtained by repeated applications of the linear operator $\mathcal{N}$. This is, however, not true for nonlinear operators.

Remark: In particular, the numerical method is stable if $||\mathcal{N}|| < 1$, since in this case, we have

$$||\mathcal{N}^n|| \leq ||\mathcal{N}||^n < 1. \tag{4.43}$$

Theorem: The Lax Equivalence Theorem for linear difference methods states that, for a well-posed consistent, linear method, stability is necessary and sufficient for convergence.

A full proof can be found in a book by Richtmyer and Morton, *Difference Methods for Initial-Value Problems*, Wiley-Interscience, 1967, and we only partially prove the sufficient part of the claim:

`consistency + stability` $\Longrightarrow$ `convergence`

Proof: We are going to show

$$\lim_{\Delta t, \Delta x \to 0} ||E^{n+1}_g|| = 0. \tag{4.44}$$

Since $\mathcal{N}$ is linear, Eq. 4.41 becomes, recursively,

$$\begin{aligned}
||E^{n+1}_g|| &\leq \Delta t||E^{n+1}_{LT}|| + ||\mathcal{N}(u^n - E^n_g) - \mathcal{N}(u^n)|| && (4.45)\\
&= \Delta t||E^{n+1}_{LT}|| + ||\mathcal{N}(E^n_g)|| && (4.46)\\
&\leq \Delta t||E^{n+1}_{LT}|| + ||\mathcal{N}||||E^n_g|| && (4.47)\\
&\leq \Delta t||E^{n+1}_{LT}|| + C||E^n_g|| && (4.48)\\
&\leq \Delta t||E^{n+1}_{LT}|| + C\Big(||\mathcal{N}||||E^{n-1}_g|| + \Delta t||E^n_{LT}||\Big) && (4.49)\\
&\cdots && (4.50)\\
&\leq \Delta t \sum_{j=1}^{n+1} C^{n+1-j}||E^j_{LT}|| + C^{n+1}||E^0_g|| && (4.51)\\
&\leq \tilde{D}(n+1)\Delta t||E_{LT}|| + \tilde{C}||E^0_g|| && (4.52)\\
&= \tilde{D}t^{n+1}||E_{LT}|| + \tilde{C}||E^0_g||, && (4.53)
\end{aligned}$$

where $||E_{LT}|| = \max_{1\le j\le n+1} ||E_{LT}^j||$, and for some $\tilde{C}$ and $\tilde{D}$.

Now if we let $\Delta t \to 0$, then $||E_g^0|| \to 0$, since it is the global error on resolving the discrete initial data. It has to go to zero when the grid gets more and more refined unless the initial data has some numerical error to start with (i.e., ill-posed problems).

Also, if we let $\Delta t \to 0$, then $||E_{LT}|| \to 0$, since the method is consistent by assumption. Therefore, we prove $||E_g^{n+1}|| \to 0$ as $\Delta x, \Delta t \to 0$, and the method is convergent.

Note: It is not hard to show that the the sufficient condition also holds when $\mathcal{N}$ is contractive, i.e.,

$$||\mathcal{N}(P) - \mathcal{N}(Q)|| \le ||P - Q||. \tag{4.54}$$

Remark: One can also say the method is stable in $||\cdot||$ if

$$||U^{n+1}|| \le ||U^n||, \tag{4.55}$$

for all n. To show this, let us assume Eq. 4.55. Recalling $U^{n+1} = \mathcal{N}(U^n)$, we have

$$\frac{||\mathcal{N}(U^n)||}{||U^n||} = \frac{||U^{n+1}||}{||U^n||} \le 1, \tag{4.56}$$

for $||U^n|| \neq 0$. Since Eq. 4.56 is true for all n, we can take sup to get

$$\sup_{U\neq 0} \frac{||\mathcal{N}(U)||}{||U||} \le 1 \tag{4.57}$$

which gives

$$||\mathcal{N}|| \le 1. \tag{4.58}$$

Hence Eq. 4.55 implies the method is stable.

Problem 3 Show that the upwind scheme of our model PDE with $a > 0$ is stable if $0 < C_a \le 1$.

Problem 4 Show that the downwind scheme of our model PDE with $a > 0$ is not stable for $0 < C_a$.

Quick summary: `Consistency + Stability` $\Longleftrightarrow$ `Convergence`

- Consistency: $\lim_{\Delta t,\Delta x\to 0} ||E_{LT}^n|| = 0$.
- Stability: $||\mathcal{N}^n|| \le C$, $\forall n\Delta t \le T$ for each T. Equivalently, $||U^{n+1}|| \le ||U^n||$.
- Convergence: $\lim_{\Delta x,\Delta t\to 0} ||E_g^n|| = 0$.

4. The CFL Condition

We can write an explicit DE for our model PDE as a conservative form

$$U_i^{n+1} = U_i^n - C_a\Big(F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}\Big), \tag{4.59}$$

where a form of the numerical fluxes can have of the form

$$F^n_{i+\frac{1}{2}} = \mathcal{F}(U_i^n, U_{i+1}^n) = \begin{cases} aU_i^n \text{ if } a > 0 \\ aU_{i+1}^n \text{ if } a < 0. \end{cases} \tag{4.60}$$

Similarly,

$$F^n_{i-\frac{1}{2}} = \mathcal{F}(U_{i-1}^n, U_i^n) = \begin{cases} aU_{i-1}^n \text{ if } a > 0 \\ aU_i^n \text{ if } a < 0. \end{cases} \tag{4.61}$$

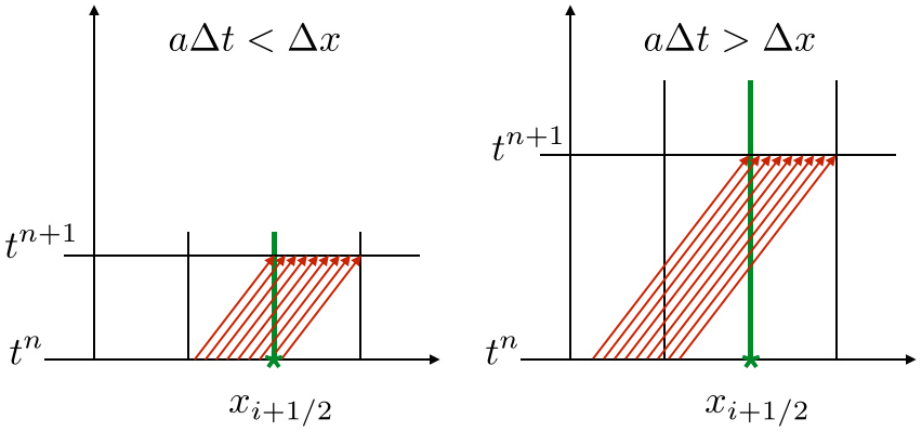


Figure 3. Characteristics for the model advection equation with $a > 0$. Left panel: For a small Δt satisfying $\Delta t < \Delta x/a$, the characteristic information travels 'less' than a single grid cell distance in a single time step Δt, hence the numerical flux $F^n_{i+\frac{1}{2}}$ at $x_{i+\frac{1}{2}}$ depends on U_i^n only. Right panel: For a large Δt failing to satisfy $\Delta t < \Delta x/a$, the characteristic information travels 'more' than a single grid cell distance in a single time step Δt, giving the extended dependency of the numerical flux $F^n_{i+\frac{1}{2}}$ on U^n_{i-1} as well as U_i^n.

Note that the method in Eq. 4.59 reduces to the stable, upwind method in Eq. A.19. In general, when the model PDE is no longer a linear constant scalar case but nonlinear systems (e.g., the Euler equation), one needs to have a conditional statement to consider signs of $a_{i\pm1/2}$ and produce the cell-interface fluxes of the form $a^n_{i\pm1}U^n_{i\pm1}$. We consider these more sophisticated cases much later when we study numerical methods for nonlinear systems and just focus on a simple linear constant scalar for now.

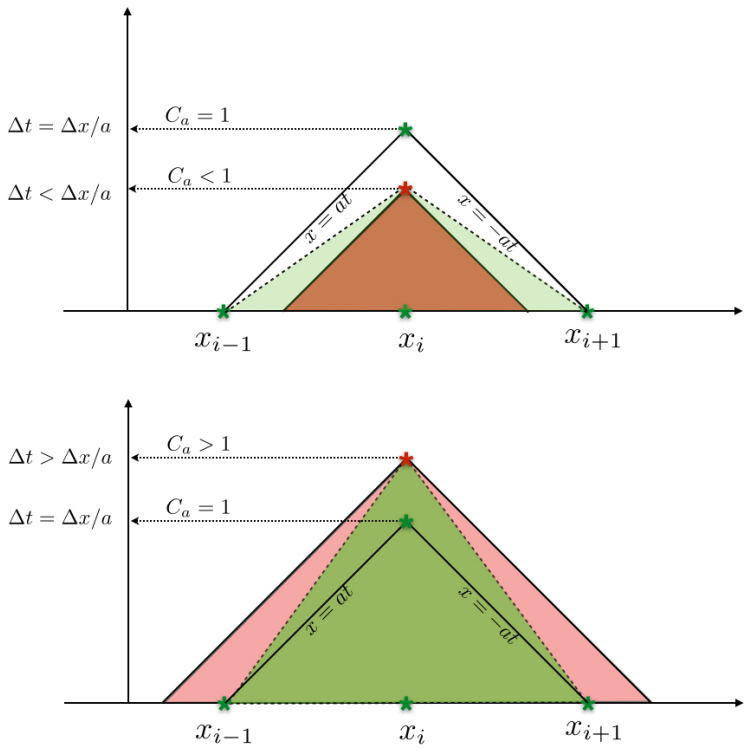


Figure 4. The green triangles represent the numerical domain of dependence of the point marked by the red stars. The red triangles illustrate the analytical domain of dependence of the red point. The solid triangles with two characteristic lines $x = \pm at$ show the maximum CFL stability regions of $C_a = 1$. Top figure: Illustration of a stable case where the numerical domain of dependence (green triangle) includes all the analytical domain (red triangle) of dependence. Bottom figure: Illustration of an unstable case where the numerical domain of dependence (green triangle) does not include all the analytical domain (red triangle) of dependence.

We see that the update scheme of DE in Eq. 4.59 uses basically three neighboring cell data, $U^n_{i-1}, U^n_i, U^n_{i+1}$ in order to update U^{n+1}_i over Δt. One can think of two different situations in choosing Δt:

$$(1) \qquad a\Delta t < \Delta x \tag{4.62}$$

$$(2) \qquad a\Delta t > \Delta x. \tag{4.63}$$

For the first case in Eq. 4.62, information propagates less than one grid cell distance in a single time step, whereas information travels much longer distance than one grid cell for the second case in Eq. 4.63.

As shown in the right panel of Fig. 3, the way we formulate the numerical flux $F^n_{i+\frac{1}{2}}$ in Eq. 4.60 would become unstable because it does not include U^n_{i-1}

for the large choice of $\Delta t > \Delta x/a$. As a result, the numerical scheme in Eq. 4.59 will become unstable in this large single time step Δt, and hence the instability will grow exponentially.

This is a consequence of the CFL condition, named after Courant, Friedrichs, and Lewy. See those three genius faces on top of the Pantheon pillars in Fig. 1 of the cover page. The CFL condition is a necessary stability condition for any numerical method and is stated as follows:

A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the given PDE, at least in the limit as Δt and Δx go to zero.

The CFL condition therefore provides a necessary condition for choosing the length of Δt depending on the PDE under consideration. The CFL condition amounts to say, if we let C_a to be the CFL number that satisfy $0 < C_a \leq 1$, C_a becomes, for the advection case,

$$C_a = \max_p |\lambda_p| \frac{\Delta t}{\Delta x}, \tag{4.64}$$

and for the diffusion case,

$$C_a = \max_p \kappa_p \frac{2\Delta t}{\Delta x^2}, \tag{4.65}$$

where p is the number of all available wave speeds λ_p or the diffusion coefficients κ_p, respectively. Note that $p = 1$ for the linear 'scalar' equations.

It is important to note that the CFL condition is only a *necessary* condition for stability (and hence convergence). It is not always *sufficient* to guarantee stability, and a numerical method satisfying the CFL condition can become unstable.

So far, we have discussed *stability* of the numerical schemes related to the CFL condition. What can we say about the numerical *accuracy* regarding to the CFL condition? Let us try to give a brief discussion about the relation between C_a and the numerical accuracy. Consider the stable case shown in the top figure in Fig. 4. We know from Chapter 2 that the properties at the red star depend only on those points inside the red triangle. However, the grid points x_{i-1} and x_{i+1} are *outside* the domain of dependence – the red triangle – for the red star and hence *theoretically* they should not influence the properties of the red star. On the other hand, the *numerical* domain of dependence – the region of the green triangle – actually takes information *only* from the two locations at x_{i-1} and x_{i+1} (the red triangle region is indeed under resolved as there is no grid point there!) which are outside the analytical domain of dependence. This means that, when Δt is chosen to be very small (i.e., $\Delta t \ll \Delta x/a$) the numerical results at the red star may be quite *inaccurate* due to the large discrepancy between the analytical domain of dependence of the red star and the location of the actual numerical data used to calculate properties at the red star.

In summary, we conclude that $C_a \leq 1$ for stability, but at the same time, it is desirable to have $C_a \approx 1$ as much as possible for accuracy. In order to have such a numerical scheme, one has to work hard to provide a better numerical methods that allow C_a to be as large as possible in a stable manner. Often, such work requires to implement *high-order, stable* numerical algorithms, especially for multi-dimensions.

Example: The CFL condition of the first-order donor-cell upwind (DCU) method in 2D has

$$\max_{x,y} \left\{ \frac{|\lambda_x|}{\Delta x} + \frac{|\lambda_y|}{\Delta y} \right\} \Delta t \leq 1, \tag{4.66}$$

where λ_x and λ_y are the maximum characteristic wave speeds in x- and y-directions. This reduces the CFL stability region in 2D by half (i.e., $C_a \leq 0.5$) compared to the 1D case.

Example: The CFL condition of the first-order donor-cell method in 3D has

$$\max_{x,y,z} \left\{ \frac{|\lambda_x|}{\Delta x} + \frac{|\lambda_y|}{\Delta y} + \frac{|\lambda_z|}{\Delta z} \right\} \Delta t \leq 1. \tag{4.67}$$

This thus reduces the CFL stability region in 3D by 1/3 (i.e., $C_a \leq 1/3$) compared to the 1D case.

Example: On the other hand, the CFL condition of the second-order corner-transport-upwind (CTU) method in 2D and 3D has

$$\max_{x,y} \left\{ \frac{|\lambda_x|}{\Delta x}, \frac{|\lambda_y|}{\Delta y} \right\} \Delta t \leq 1, \tag{4.68}$$

$$\max_{x,y,z} \left\{ \frac{|\lambda_x|}{\Delta x}, \frac{|\lambda_y|}{\Delta y}, \frac{|\lambda_z|}{\Delta z} \right\} \Delta t \leq 1, \tag{4.69}$$

$$\tag{4.70}$$

respectively. The CTU method therefore keeps the same CFL stability region in multi-dimensions as in the 1D case, $C_a \leq 1$. Implementing CTU is much more complicated than DC.

5. Stability using von Neumann Analysis

For constant-coefficient linear equation, von Neumann stability analysis is often the easiest way to determine stability bounds. As briefly mentioned in Section 2., the use of l^2-norm becomes particularly useful because of Parseval's relation,

$$||U^n||_2 = ||\hat{U}^n||_2. \tag{4.71}$$

The idea to show the boundedness of U^n is then equivalent to showing the boundedness of $\hat{U}^n$. This is convenient because each Fourier mode of $\hat{U}^n$ can be decoupled from all other wave numbers, whereas all elements of U^n are coupled

together via the difference equations. In fact, the Fourier transform diagonalizes the linear difference operator and hence we can easily decouple each individual Fourier mode by transforming U^n in the real space to $\hat{U}^n$ in the Fourier space.

This brings us an easy analysis tool in Fourier space and it suffices to consider an arbitrary single wave number ξ and the data of the form

$$U_j^n = A(t^n)e^{I\xi j\Delta x}, \tag{4.72}$$

where $I = \sqrt{-1}$ and $A(t^n) = e^{\alpha n\Delta t}$ is a complex number ($\alpha \in \mathbb{C}, n \in \mathbb{Z}, \Delta t \in \mathbb{R}$) called the amplification factor. We now use the Fourier analysis to obtain the CFL condition $0 < C_a \leq 1$ once again – this Fourier analysis is often called the von Neumann stability analysis for constant-coefficient linear models.

Let's again take our model DE for the 1D advection with $a > 0$,

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{\Delta x}\Big(U_j^n - U_{j-1}^n\Big). \tag{4.73}$$

Substituting Eq. 4.72 into Eq. 4.73, we have

$$e^{\alpha(n+1)\Delta t}e^{I\xi j\Delta x} = e^{\alpha n\Delta t}e^{I\xi j\Delta x} - e^{\alpha n\Delta t}C_a\Big[e^{I\xi j\Delta x} - e^{I\xi(j-1)\Delta x}\Big]. \tag{4.74}$$

Dividing Eq. 4.74 by $e^{\alpha n\Delta t}e^{I\xi j\Delta x}$, we get

$$\begin{aligned} e^{\alpha\Delta t} &= 1 - C_a\Big[1 - e^{-I\xi\Delta x}\Big] && (4.75)\\ &= 1 - C_a + C_a\cos(\xi\Delta x) + IC_a\sin(\xi\Delta x). && (4.76)\end{aligned}$$

Recalling from the relation 4.55 for numerical stability $||U^{n+1}|| \leq ||U^n||$, we see that it suffices to show the amplification factor is bounded by unity:

$$|e^{\alpha\Delta t}| \leq 1. \tag{4.77}$$

Therefore, we consider

$$\begin{aligned} 1 \geq |e^{\alpha\Delta t}|^2 &= \Big(1 - C_a + C_a\cos(\xi\Delta x)\Big)^2 + C_a^2\sin(\xi\Delta x) && (4.78)\\ &= 1 - 2C_a(1 - C_a)\Big(1 - \cos(\xi\Delta x)\Big). && (4.79)\end{aligned}$$

Since $1 - \cos(\xi\Delta x) \geq 0$, we get

$$C_a(1 - C_a) \geq 0, \tag{4.80}$$

which gives

$$0 \leq C_a \leq 1. \tag{4.81}$$

Problem 5 Repeat the similar von Neumann analysis of the 1D advection using forward in time forward in space (FTFS)

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{\Delta x}\Big(U_{j+1}^n - U_j^n\Big). \tag{4.82}$$

Conclude that FTFS is unstable if $a > 0$ and stable if $a < 0$.

Problem 6 Use von Neumann stability analysis to show that the CFL condition for the 1D heat equation

$$U_j^{n+1} = U_j^n + C_a\Big(U_{j+1}^n - 2U_j^n + U_{j-1}^n\Big), \tag{4.83}$$

with $C_a = \kappa\frac{2\Delta t}{\Delta x^2}$, becomes $C_a \leq 1$ as before.

Problem 7 Show that a forward in time centered in space scheme (FTCS) for 1D advection with $a > 0$

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}\Big(U_{j+1}^n - U_{j-1}^n\Big) \tag{4.84}$$

is unconditionally unstable.

Problem 8 Show that an implicit scheme of backward in time centered in space (BTCS)

$$U_j^{n+1} = U_j^n - \frac{a\Delta t}{2\Delta x}\Big(U_{j+1}^{n+1} - U_{j-1}^{n+1}\Big) \tag{4.85}$$

is unconditionally stable.

Problem 9 Bonus Problem What does von Neumann stability analysis say about the LF method?

6. A List of Finite Difference Methods for the Linear Problem

In this section, we provide a couple of finite difference (FD) methods for solving our model PDE, $u_t + au_x = 0$. We assume $a > 0$ for Beam-Warming and Fromm's methods. One can easily get appropriate forms for these two methods for $a < 0$.

- Backward Euler (FTCS – Forward Time Centered Space)

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{2\Delta x}\Big(U_{i+1}^n - U_{i-1}^n\Big) \tag{4.86}$$

- One-sided (FTBS – Forward Time Backward Space)

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{\Delta x}\Big(U_i^n - U_{i-1}^n\Big) \tag{4.87}$$

- One-sided (FTFS – Forward Time Forward Space)

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{\Delta x}\Big(U_{i+1}^n - U_i^n\Big) \tag{4.88}$$

- Leapfrog

$$U_i^{n+1} = U_i^{n-1} - \frac{a\Delta t}{\Delta x}\Big(U_{i+1}^n - U_{i-1}^n\Big) \tag{4.89}$$

- Lax-Friedrichs (LF)

$$U_i^{n+1} = \frac{1}{2}\Big(U_{i+1}^n + U_{i-1}^n\Big) - \frac{a\Delta t}{2\Delta x}\Big(U_{i+1}^n - U_{i-1}^n\Big) \tag{4.90}$$

- Lax-Wendroff (LW)

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{2\Delta x}\Big(U_{i+1}^n - U_{i-1}^n\Big) + \frac{1}{2}\Big(\frac{a\Delta t}{\Delta x}\Big)^2\Big(U_{i+1}^n - 2U_i^n + U_{i-1}^n\Big) \tag{4.91}$$

- Beam-Warming (BW) for $a > 0$

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{2\Delta x}\Big(3U_i^n - 4U_{i-1}^n + U_{i-2}^n\Big) + \frac{1}{2}\Big(\frac{a\Delta t}{\Delta x}\Big)^2\Big(U_i^n - 2U_{i-1}^n + U_{i-2}^n\Big) \tag{4.92}$$

- Fromm's method for $a > 0$

$$\begin{aligned} U_i^{n+1} = \quad & U_i^n - \frac{a\Delta t}{\Delta x}\Big(U_i^n - U_{i-1}^n\Big) - \frac{1}{4}\frac{a\Delta t}{\Delta x}\Big(1 - \frac{a\Delta t}{\Delta x}\Big)\Big(U_{i+1}^n - U_i^n\Big) \\ & + \frac{1}{4}\frac{a\Delta t}{\Delta x}\Big(1 - \frac{a\Delta t}{\Delta x}\Big)\Big(U_{i-1}^n - U_{i-2}^n\Big) \end{aligned} \tag{4.93}$$

Chapter 5

Computing Discontinuous Solutions of Linear Conservation Laws

For conservation laws we often encounters discontinuous solutions, and handling discontinuities successfully in numerical methods is indeed our great interests. So far, we have studied the linear theory where we naturally assumed smooth solutions especially in our discussion of the truncation error and convergence theory.

We now wish to understand what would happen when we apply a numerical method that is proven to work well for smooth solutions to a linear problem with discontinuous initial data.

1. Non-convergence at Discontinuity

In order to study numerical solution behaviors near discontinuity, let us start considering the scalar advection equation with the Riemann problem as its initial condition:

$$u_t + au_x = 0, x \in \mathbb{R}, t \geq 0 \tag{5.1}$$

$$u_0(x) = \begin{cases} 1 \text{ for } x < 0 \\ 0 \text{ for } x > 0. \end{cases} \tag{5.2}$$

We know that the exact solution is $u_0(x-at)$ which is easily achieved by tracing the characteristic curve. Does this solution behave well at the discontinuity? To address this issue, let's consider what happens to numerically compute the spatial derivative u_x across the discontinuity. For $t > 0$, the shock travels to the location $x = at$ from its initial location $x = 0$. Letting $x_1 = at + \Delta x$ and $x_2 = at - \Delta x$, a finite difference approximation to u_x applied across the discontinuity at $x = at$ becomes

$$\left.\frac{\partial u}{\partial x}\right|_{x=at} = \frac{u(x_1,t) - u(x_2,t)}{x_1 - x_2} \tag{5.3}$$

$$= \frac{u_0(\Delta x, t) - u_0(-\Delta x, t)}{2\Delta x} \tag{5.4}$$

$$= \frac{0-1}{2\Delta x} \to -\infty \tag{5.5}$$

as $\Delta x \to 0$.

The local truncation error does not vanish as $\Delta x \to 0$ and the method becomes inconsistent, therefore the proof of convergence we studied in the previous chapter is no longer available.

We can rescue the failure of the convergence proof by adopting the vanishing viscosity approximation $u_0^\epsilon(x)$ to $u_0(x)$ and letting $\epsilon \to 0$. Although this vanishing viscosity approximation can resurrect the convergence of the method, the *convergence rate* may be severely dropped and lower than the 'order' of the method as defined on smooth solutions. The outcome of such numerical difficulties involving discontinuity would make its numerical solutions look very unsatisfactory on any particular finite grid.

In Fig. 1 we display five different numerical solutions to two different types of initial conditions. The panels on the left column shows the smooth $\sin(2\pi x)$ wave initialized on $x \in [0, 1]$. The sine wave is solved numerically with – from top to bottom – (1) Upwind method, (2) Lax-Friedrichs, (3) Lax-Wendroff, (4) Beam-Warming, and (5) Fromm's method. On the right column, the same methods are applied – in the same order – to solve the initially discontinuous Riemann problem,

$$u_0(x) = \begin{cases} 1 \text{ for } x < 0.5 \\ -1 \text{ for } x > 0.5. \end{cases} \tag{5.6}$$

All numerical methods solve the sine wave until the wave completes the first cycle on a periodic domain which is resolved on 64 grid cells, $N = 64$. The same number of grid cells is used for the discontinuous case where the solutions have been integrated on a domain with outflow boundary condition until the location of the shock reaches to $x = 0.8$ which is 0.3 distance away from its initial location $x = 0.5$.

There are two first-order methods (upwind and Lax-Friedrichs) and three second-order methods (Lax-Wendroff, Beam-Warming, and Fromm's method). We note that all methods behave equally well on the smooth flow. On the contrary, there are two distinctive solution characteristics – dissipation and oscillations – on the discontinuous flow, particularly near the discontinuity: the first-order methods give very smeared solutions, while the second-order methods give oscillations. These types of behaviors are very typical and understanding the origin of such numerical behaviors is our goal in this chapter. Once we understand the fundamental sources of the issues we would be able to improve them and provide better numerical algorithms.

In addition, if we compute the l^1-norm error in these computed discontinuous solutions we do not see the expected rates of convergence that would be expected on smooth solutions. It turns out that the first-order methods converge with an error that is $\mathcal{O}(\Delta x^{1/2})$ while the second-order methods have an

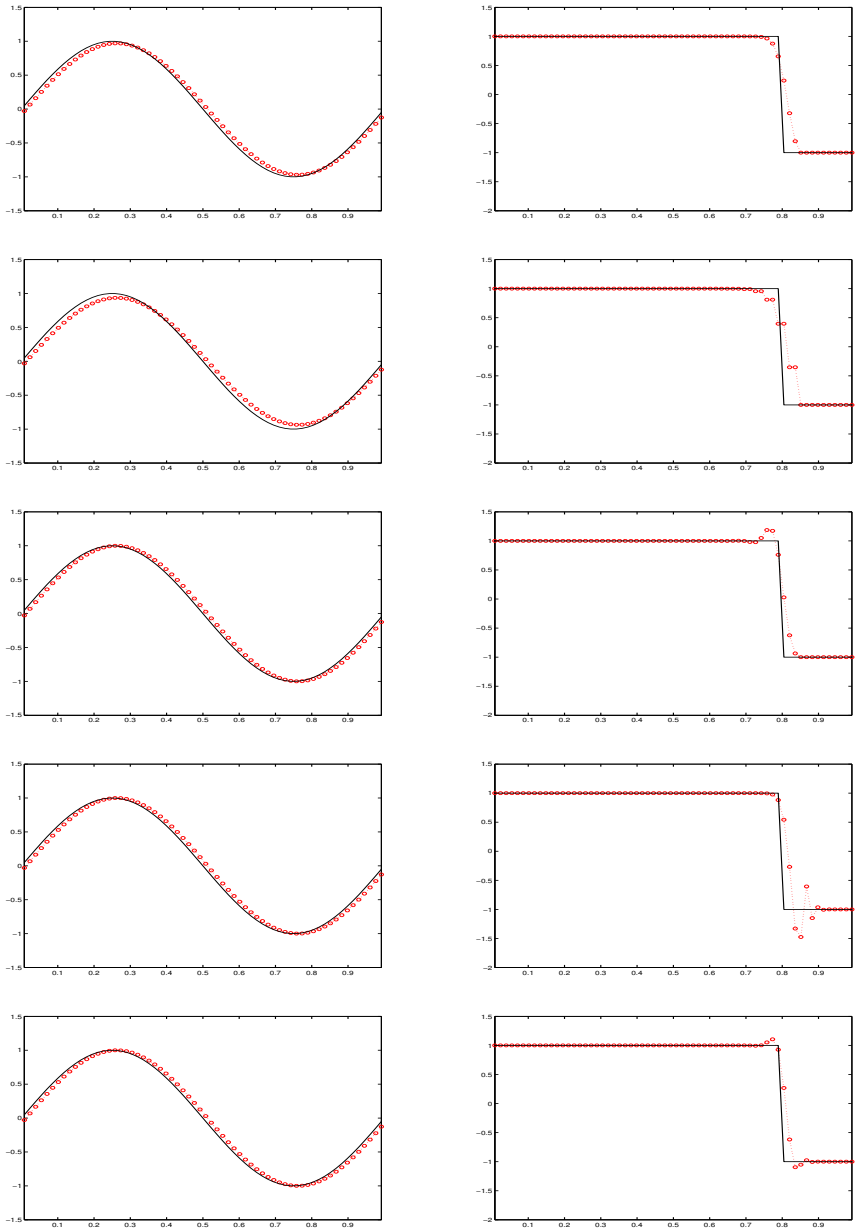


Figure 1. Numerical (red circles) and exact (black solid curves) solutions to the scalar advection equation $u_t + au_x = 0, a > 0$ with two different initial conditions: Left column: sinusoidal wave, Right column: discontinuous Riemann problem. Five different schemes are shown from top to bottom: (1) Upwind, (2) Lax-Friedrichs, (3) Lax-Wendroff, (4) Beam-Warming, (5) Fromm's method.

error that is $\mathcal{O}(\Delta x^{2/3})$ at best. These convergence rates can be proved for very general initial data using the vanishing viscosity approximations.

2. Modified Equations

Recall that we have followed the following two-step procedure all the time so far in order to discuss theories on the numerical PDEs:

1. First, identify PDEs to solve numerically, and
2. Second, discretize PDEs to get approximate solutions using relevant DEs.

We now try to reverse the process:

1. First, start with DEs, and
2. Second, consider relevant PDEs associated with the given DEs.

At first glance, it seems bit strange why we would want to do this, since the reverse process may lead us to go back to the original PDEs we start from in the conventional process. However, it turns out that the reverse procedure provides us with a very useful information for studying the solution behaviors of DEs via modeling the DEs by PDEs. This also means that the end product of the reverse process is *not* the original PDEs, but different PDEs with extra information, which is called the *modified equations*. The extra information enables us to understand the 'qualitative' behavior of the numerical methods we discussed in the previous section (see also Fig. 1) such as

- dissipative (or diffusive) solution behaviors across discontinuities in the first-order methods (e.g., upwind, Lax-Friedrichs, etc.), and
- oscillatory (or dispersive) solution behaviors near discontinuities in the second-order methods (e.g., Lax-Wendroff, Beam-Warming, Fromm's method, etc.).

The derivation of the modified equation is closely related to the calculation of the local truncation error $E^n_{LT,i}$ in Eq. 4.16 for a given DE. We are going to see that there are two types of numerical relations, dissipation and dispersion, that can be derived from the modified equation approach. The dissipation relation is an outcome of identifying a modified equation for odd-order accurate schemes (e.g., first-, third-order methods), while the dispersion relation is available from looking at a modified equation for even-order accurate schemes (e.g., second-, fourth-order methods). In the subsequent sections, we consider set of modified equations for first- and second-order methods.

2.1. Dissipation Error in First-order Methods

We start with the first-order upwind method (FTBS) for the linear scalar advection $u_t + au_x = 0$ with $a > 0$:

$$U_i^{n+1} = U_i^n - \frac{a\Delta t}{\Delta x}\Big(U_i^n - U_{i-1}^n\Big). \tag{5.7}$$

The analysis of obtaining a modified equation is very closely related to computing the local truncation error $E^n_{LT,i}$ as in Eq. 4.24, where we have assumed

that $u(x,t)$ is an analytical solution to $u_t + au_x = 0$.

This time, however, we no longer make the same assumption on $u(x,t)$, but instead we make the following assumption:

`Assume that` $u(x,t)$ `exactly agrees with` U_i^n `at the discrete grid points, or mathematically,` $u(x_i,t^n) = U(x_i,t^n)$.

In this way, $u(x,t)$ satisfies the DE relation Eq. 5.7 exactly:

$$u(x,t+\Delta t) = u(x,t) - \frac{a\Delta t}{\Delta x}\Big(u(x,t) - u(x-\Delta x,t)\Big). \tag{5.8}$$

Expanding these terms in Taylor series around (x,t) and simplifying gives:

$$\Big(u_t + \frac{\Delta t}{2}u_{tt} + \frac{\Delta t^2}{6}u_{ttt} + \cdots\Big) + a\Big(u_x - \frac{\Delta x}{2}u_{xx} + \frac{\Delta x^2}{6}u_{xxx} + \cdots\Big) = 0. \tag{5.9}$$

Rewriting this gives us

$$u_t + au_x = \frac{1}{2}\Big(au_{xx}\Delta x - u_{tt}\Delta t\Big) - \frac{1}{6}\Big(au_{xxx}\Delta x^2 + u_{ttt}\Delta t^2\Big)\cdots \tag{5.10}$$

Noticing we have $u_{tt} = a^2u_{xx} + \mathcal{O}(\Delta t)$ (note that we no longer have $u_{tt} = a^2u_{xx}$ since $u(x,t)$ is not an exact solution of $u_t + au_x = 0$), we can obtain

$$\begin{aligned} u_t + au_x &= \frac{1}{2}\Big(au_{xx}\Delta x - a^2u_{xx}\Delta t\Big) + \mathcal{O}(\Delta x^2, \Delta t^2) && (5.11)\\ &= \frac{a\Delta x}{2}\Big(1 - a\frac{\Delta t}{\Delta x}\Big)u_{xx} + \mathcal{O}(\Delta x^2, \Delta t^2) && (5.12)\end{aligned}$$

Dropping $\mathcal{O}\Delta x^2, \Delta t^2$ terms, we notice that the above equation is an advection-diffusion equation of the form

$$u_t + au_x = \kappa u_{xx}, \tag{5.13}$$

with a constant diffusion coefficient κ. Similarly, we can proceed to compute a diffusion coefficient for LF (see **Homework 1**), and we have:

$$\kappa = \begin{cases} \frac{a\Delta x}{2}\Big(1 - C_a\Big) \text{ for upwind,} \\ \\ \frac{\Delta x^2}{2\Delta t}\Big(1 - C_a^2\Big) \text{ for LF.} \end{cases} \tag{5.14}$$

Remark: In the analysis done in Eq. 4.24, we used the fact that $u(x,t)$ is the true solution of the linear scalar advection equation, henceforth we made use of the fact $u_t + au_x = 0$ in the $E_{LT,i}^n$ derivation in Eq. 4.24, giving the local truncation error is of first-order in space and time:

$$E_{LT,i}^{n+1} \approx \mathcal{O}(\Delta t, \Delta x). \tag{5.15}$$

However, this time, if we instead take $u(x,t)$ to be the solution of the PDE

$$u_t + au_x = \frac{a\Delta x}{2}\Big(1 - \frac{a\Delta t}{\Delta x}\Big)u_{xx}, \tag{5.16}$$

the truncation error would be of second-order accurate

$$E^{n+1}_{LT,i} \approx \mathcal{O}(\Delta t^2, \Delta x^2), \tag{5.17}$$

and consequently, the upwind DE method is also second-order accurate when approximating the modified equation Eq. 5.16.

Note what we just have done so far: we first took our model DE and use Taylor expansions to arrive at a new PDE that our DE actually solves for. As clearly seen from a variant form of the modified equation that is different from the original advection PDE, we see that our DE is *not* solving the original PDE $u_t + au_x = 0$, but rather it solves the modified equation.

This is why we see the numerical solutions from the first-order methods become diffusive (or smeared out) as time evolves. The reason is clear now that the extra diffusion term in the modified equation become active and add numerical diffusion across discontinuities. See Fig. 1. If we simply plot the *exact* solutions to the modified equations in Eq. 5.16 and Eq. 5.19 together with the numerical solutions of the upwind and LF solutions, they are virtually indistinguishable to plotting accuracy.

Note that the diffusive term is of order $\mathcal{O}(\Delta x)$ as $\Delta x \to 0$ and hence it vanishes in the limit. This means that the numerical solutions produced by the upwind and LF methods are indeed very good approximations to the vanishing viscosity solutions of the two methods, u^ϵ. In the linear case there is only one weak solution to which u^ϵ converges, while for nonlinear cases, it turns out that the LF method satisfies a discrete entropy condition and converges more generally to the vanishing viscosity weak solution as the grid is refined.

In real calculation, one can also compare the magnitude of the two diffusion coefficients κ in Eq. 5.14 and determine which is more diffusive. For instance, if we choose Δt so that it satisfies $C_a = 0.8$ on a given grid resolution (e.g., $\Delta t = 0.8, \Delta x = 1.0, a = 1.0$), we see from Eq. 5.14 that

$$\kappa = \begin{cases} 0.1000 \text{ for upwind,} \\ 0.1152 \text{ for LF.} \end{cases} \tag{5.18}$$

Therefore, the Lax-Friedrichs method is more diffusive than the upwind method.

In Fig. 1, it is shown that the LF solution also exhibits a very distinctive phenomenon called "odd-even decoupling" in which the numerical solution experiences spurious oscillations (which is different from the *fluctuating* oscillations in the second-order methods) with the shortest possible wave length of $2\Delta x$.

Problem 1 Show that a modified equation for Lax-Friedrichs method is

$$u_t + au_x = \frac{\Delta x^2}{2\Delta t}\Big(1 - C_a^2\Big)u_{xx}, \tag{5.19}$$

where $C_a = \frac{a\Delta t}{\Delta x}$.

Remark: For stability, we know the diffusion coefficient needs be positive $\kappa > 0$ all the time in the diffusion equation (see also Table 1 in Chapter 2). Using this fact, we can easily retrieve the CFL condition, $0 \leq C_a \leq 1$, once again for the upwind and LF methods (assuming $a > 0$) from their modified equations Eq. 5.16 and Eq. 5.19, respectively.

Quick summary:

- In this section we obtained two modified equations of the two first-order DEs (upwind and LF), by expanding Taylor series using the original PDEs.
- We see that new extra information appears as a diffusive term.
- This explains why the first-order methods experience diffusive behavior across discontinuities.
- We refer this phenomenon the 'dissipation error (or diffusion error)' in first-order methods.

Problem 2 Consider

$$U_j^{n+1} = U_j^n - \frac{C_a}{2}\Big(U_{j+1}^n - U_{j-1}^n\Big), \tag{5.20}$$

and explain why that the method is unstable for all $\Delta x/\Delta t$ by using the modified equation analysis.

2.2. Dispersion Error in Second-order Methods

For our second-order methods, we consider the Lax-Wendroff (LW) method and the Beam-Warming (BW) method on $u_t + au_x = 0$. We can see that the modified equations for the two methods are, respectively,

$$u_t + au_x = \frac{a\Delta x^2}{6}\Big(C_a^2 - 1\Big)u_{xxx} \text{ for LW}, \tag{5.21}$$

$$u_t + au_x = \frac{a\Delta x^2}{6}\Big(C_a^2 - 3C_a + 2\Big)u_{xxx} \text{ for BW}. \tag{5.22}$$

Both of these modified equations have the form

$$u_t + au_x = \mu u_{xxx}, \tag{5.23}$$

which is a *dispersion* equation with a dispersion coefficient μ:

$$\mu = \begin{cases} \frac{a\Delta x^2}{6}\left(C_a^2 - 1\right) \text{ for LW,} \\ \frac{a\Delta x^2}{6}\left(C_a^2 - 3C_a + 2\right) \text{ for BW.} \end{cases} \tag{5.24}$$

Problem 3 Show Eq. 5.21 and 5.22.

The theory of dispersive waves can be easily understood if we take a look at a Fourier series solution to this equation. Recall that we can write $u(x,t)$ using the Fourier component $\hat{u}(\xi,t)$, and vice versa:

$$u(x,t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{u}(\xi,t) e^{I\xi x} d\xi, \tag{5.25}$$

$$\hat{u}(\xi,t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} u(x,t) e^{-I\xi x} dx, \tag{5.26}$$

where ξ is a wave number and $I = \sqrt{-1}$.

Note: In this way, we can say that the Fourier components with different wave number ξ propagate at different speeds, i.e., they disperse as time evolves. And by linearity it suffices to consider each wave number in isolation – one big attractive thing you can do using Fourier analysis.

Taking time and spatial derivatives of $u(x,t)$ in Eq. 5.25, we get

$$u_t(x,t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{u}_t(\xi,t) e^{I\xi x} d\xi, \tag{5.27}$$

$$u_x(x,t) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{u}_x(\xi,t) I\xi e^{I\xi x} d\xi. \tag{5.28}$$

Therefore, we easily see that the Fourier transform of u_t is $\hat{u}_t$ (trivial!), and the Fourier transform of u_x is $I\xi\hat{u}$:

$$\hat{u}_t = \hat{u}_t, \text{ and } \hat{u}_x = I\xi\hat{u} \tag{5.29}$$

Let us now take a Fourier transform of the dispersion equation Eq. 5.23, and get

$$\hat{u}_t + aI\xi\hat{u} = \mu(I\xi)^3\hat{u}, \tag{5.30}$$

which gives

$$\hat{u}_t = -I(a\xi + u\xi^3)\hat{u} \equiv -I\omega\hat{u}. \tag{5.31}$$

Henceforth, this leads us to have

$$\hat{u}(\xi,t) = e^{-I\omega t}\hat{\eta}(\xi), \tag{5.32}$$

where $u(x,0) = \eta(x)$. One thing to notice here is that this has a character similar to advection problems in that $|\hat{u}(\xi,t)| = |\hat{\eta}(\xi)|$ for all t and each Fourier

component maintains its original amplitude – which is good.

Now some bad news. If we recombine with the inverse Fourier transform we obtain

$$\begin{aligned} u(x,t) &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{u}(\xi,t) e^{I\xi x} d\xi, & (5.33) \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{-I\omega t} \hat{\eta}(\xi) e^{I\xi x} d\xi, & (5.34) \\ &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} e^{I\xi(x-\frac{\omega}{\xi}t)} \hat{\eta}(\xi) d\xi. & (5.35) \end{aligned}$$

This indicates that the speed at which this oscillating wave propagates is clearly $\omega(\xi)/\xi$, which is called the *phase velocity* $c_p(\xi)$ for wave number ξ. This is the speed at which wave peaks travel. Consider c_p in our model case

$$c_p(\xi) = \frac{\omega(\xi)}{\xi} = a + \mu\xi^2. \tag{5.36}$$

This indicates that c_p varies with ξ and is very close to the propagation velocity a of the original advection equation only for small values of ξ. This is why we see in second-order methods those oscillatory waves at different wave numbers ξ that are traveling at different phase velocity c_p. The waves generate trains of oscillations before and after the discontinuity as shown in Fig. 1.

There is another important quantity called the *group velocity*, denoted by c_g and defined by $c_g(\xi) = \omega'(\xi)$. For our model problem, we get

$$c_g(\xi) = \frac{d\omega}{d\xi} = a + 3\mu\xi^2. \tag{5.37}$$

This is a velocity at which 'the wave packet' as a group propagates, carrying the energy associated with the overall propagation velocity. The usefulness of c_g is that it provides a good measure for general data which is usually composed of many wave numbers, therefore, different waves traveling at different phase velocities generate many 'ripples' instead of a single Gaussian wave. It is apparent that the wave length λ of the ripples is changing through this wave and the energy associated with the low wave number (i.e., larger ripples) is apparently moving faster than the energy associated with the high wave numbers (smaller ripples). The propagation velocity of this energy is the group velocity, rather than the phase velocity.

From Eq. 5.24, we can compute values of the dispersion coefficients for LW and BW. Since $0 \leq C_a < 1$ for stability, we get

$$\mu = \begin{cases} \frac{a\Delta x^2}{6}\left(C_a^2 - 1\right) < 0 \text{ for LW}, \\ \\ \frac{a\Delta x^2}{6}\left(C_a^2 - 3C_a + 2\right) > 0 \text{ for BW}. \end{cases} \tag{5.38}$$

Therefore, if we further assume $a > 0$, we get

$$c_g = \begin{cases} a + 3\mu\xi^2 < a \text{ for LW}, \\ \\ a + 3\mu\xi^2 > a \text{ for BW}. \end{cases} \tag{5.39}$$

This now explains for LW that all wave numbers travel slower than a, leading to an oscillatory wave train lagging *behind* the discontinuity as in Fig. 1. On the other hand, for BW, the oscillations are faster than the advection velocity a, resulting in the oscillations *ahead* of the discontinuity.

Remark: Please take a look at Wikipedia for a nice illustrative description on the phase velocity and the group velocity:

`http://en.wikipedia.org/wiki/Phase_velocity`

Quick summary:

- We started our discussion from the two second-order DEs,
- The modified equation approach revealed that there is an extra term called the dispersion term
- The Fourier analysis applied to the modified dispersion equations enabled us to explain the oscillatory behaviors due to the dispersion term, either ahead or behind the discontinuity.

Chapter 6

Computing Discontinuous Solutions of Non-linear Conservation Laws

In Chapter 6, we have seen in the linear conservation laws there are difficulties in computing numerical solutions at discontinuities. We now study in this chapter what happens when solving nonlinear conservation laws numerically. For nonlinear problems, there are additional difficulties that can arise:

- The discrete method might be "nonlinearly unstable", i.e., unstable on the nonlinear problem even though linearized version appear to be stable. Often oscillations will trigger nonlinear instabilities.
- The discrete method might converge to a function that is not a weak solution of our original equation, or
- The discrete method might converge to a *wrong* weak solution that does not satisfy the entropy condition.

The last case indicating we might get the wrong weak solution is not so surprising, since we already have discussed there could be infinitely many different weak solutions violating the entropy condition.

The fact pointed out in the second case, however, sounds bit more puzzling, but can be easily explained in the following way. Recall that from our previous homework problem (see Eq. 6.26 in Chapter 2), we have seen that one can derive two different weak solutions for Burgers' equation, one for

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \tag{6.1}$$

and another for

$$(u^2)_t + \left(\frac{2u^3}{3}\right)_x = 0. \tag{6.2}$$

The two equations have exactly the same smooth solutions, but the Rankine-Hunogiot condition gives different shock speeds, and hence two different weak

solutions. Let's now say that we have a finite difference method that is consistent with one of these equations, say Eq. 6.1. Then the method is also consistent with Eq. 6.2 because the Taylor series expansion – which assumes smoothness – should give the exact same result in either case, since both are exactly same for smooth solutions. We then see that although the method *is* consistent with both Eq. 6.1 and Eq. 6.2, it can only possibly converge to *one* of the two different weak solutions, say Eq. 6.2, which is *not* a weak solution of the original Burgers' equation.

Example: Let us write Burgers' equation in a nonconservative form,

$$u_t + uu_x = 0. \tag{6.3}$$

A straightforward extension of the upwind method for $u_t + au_x = 0$ can give us a natural discretization of Eq. 6.3:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} U_i^n \Big(U_i^n - U_{i-1}^n \Big), \tag{6.4}$$

where we assume $U_i^n \geq 0$ for all n and i. It turns out that the method Eq. 6.4 is adequate for smooth solutions, but does not, in general, converge to a discontinuous weak solution of Burgers' equation Eq. 6.1 as the grid is refined. For instance, consider the initial condition

$$u_0(x) = \begin{cases} 1 \text{ for } x < 0 \\ 0 \text{ for } x \geq 0. \end{cases} \tag{6.5}$$

In discrete form, we obtain

$$U_i^0 = \begin{cases} 1 \text{ for } x_i < 0 \\ 0 \text{ for } x_i \geq 0. \end{cases} \tag{6.6}$$

It is easy to check from Eq. 6.4 that $U_i^1 = U_i^0$ for all i, and recursively, we get $U_i^n = U_i^0$ for all i and n, regardless of the grid and time resolutions Δx and Δt. As the grid is refined, the numerical solution then converges very nicely to its initial function $u(x,t) = u_0(x)$ which is *not* a weak solution of Eq. 6.1 or Eq. 6.2 either.

Note: Notice that $u(x,t) = u_0(x)$ satisfying Eq. 6.5 is not a weak solution of Burgers' equation.

In this example, the solution $u(x,t) = u_0(x)$ is obviously wrong as it simply represents a standing discontinuous data for all $t > 0$, albeit the discontinuity should advect. How does the method Eq. 6.4 behave with other initial condition? We take a look at this in the next section.

1. Conservative vs. Non-conservative Schemes at Discontinuity

We now consider an initial Riemann data on $[0,1]$ given by

$$u_0(x) = \begin{cases} 2 \text{ for } x < 0.3 \\ 1 \text{ for } x \geq 0.3. \end{cases} \tag{6.7}$$

If this initial data is solved using the method in Eq. 6.4, the solution may look reasonably correct, but with a wrong propagation speed. Fig. 1 shows the true (black solid curve) and computed solution (red dotted curve) at time $t = 0.2390625$ (or after the discontinuity propagates a distance of $d = 0.35$). We can see that the solution does look very nice, but it obviously has the wrong propagation speed.

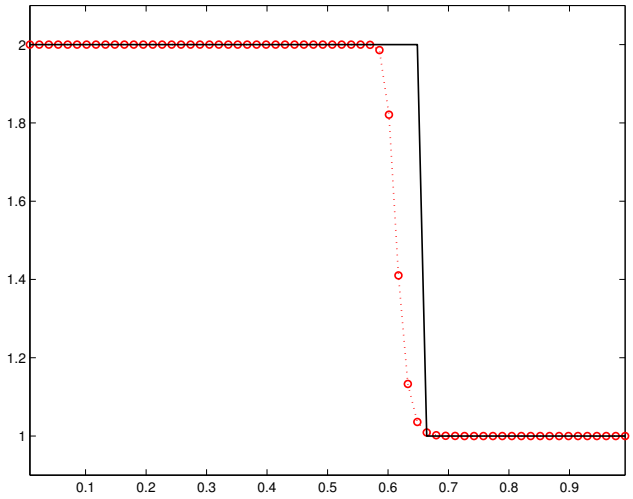


Figure 1. Numerical (red circles with dotted curve) and exact (black solid curves) solutions to Burgers' equation at $t = 0.2390625$ using the nonconservative discretization in Eq. 6.4 with the initial condition given by Eq. 6.7. The location of the discontinuity is not correct, indicating the propagation speed of the method is wrong. The numerical solution is resolved on 64 cells with $C_a = 0.9$.

The fact that we get a wrong shock propagation speed is because the method

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} U_i^n \Big(U_i^n - U_{i-1}^n \Big), \tag{6.8}$$

is not conservative.

On the other hand, let us consider the conservative discrete upwind method (a.k.a. Godunov's method) to solve the same initial data. The conservative Godunov's method takes the form

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x} \Big(\frac{1}{2} (U_i^n)^2 - \frac{1}{2} (U_{i-1}^n)^2 \Big), \tag{6.9}$$

On smooth solutions, both of these methods are first-order accurate, and they give comparable results. As shown in Fig. 1, however, the nonconservative

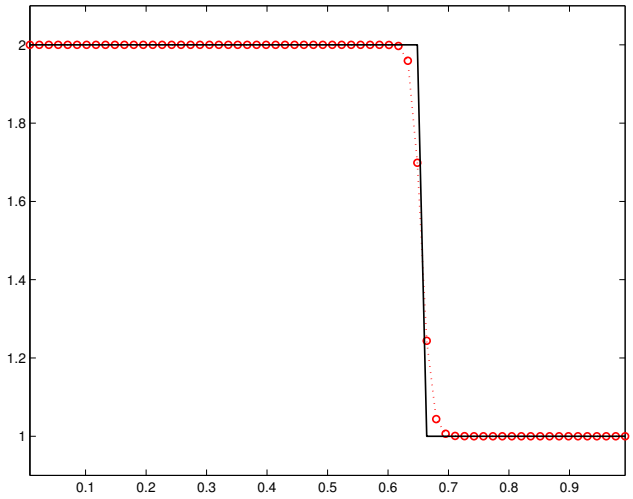


Figure 2. Numerical (red circles with dotted curve) and exact (black solid curves) solutions to Burgers' equation at $t = 0.2390625$ using the conservative discretization in Eq. 6.9 with the initial condition given by Eq. 6.7. The location of the discontinuity is correct now and the solution becomes smeared out at the discontinuity as seen in the previous chapter. The numerical solution is resolved on 64 cells with $C_a = 0.9$.

update fails to converge to a correct weak solution of the conservation law. The conservative method in Eq. 6.9 produces a slightly smeared approximation to the shock, but it is smeared about the correct location.

Note that the nonconservative scheme Eq. 6.4 can be rewritten as

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\Big(\frac{1}{2}(U_i^n)^2 - \frac{1}{2}(U_{i-1}^n)^2\Big) - \frac{1}{2}\Delta t \Delta x \Big(\frac{U_i^n - U_{i-1}^n}{\Delta x}\Big)^2. \qquad (6.10)$$

It is identical to the conservative Godunov's method except for the last term which approximates the time integral of $\frac{1}{2}\Delta x (u_x)^2$. For smooth regions, the last term is bounded and can be expected to vanish as $\Delta x \to 0$. At discontinuity, it does not vanish, and give a finite contribution in the limit, leading to a different shock speed.

The last term can be also viewed as a singular source term that is being added to the conservation law, an approximation to a delta function concentrated at the shock discontinuity.

It is worth to discuss conservation property of finite volume methods in a discrete sense. Consider a 1D domain $[a, b]$ subdivided into N subdomains, $[x_{i-1/2}, x_{i+1/2}]$ such that $[a, b] = \cup_{i=1}^{N} [x_{i-1/2}, x_{i+1/2}]$.

A conservative finite volume method takes the form

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\Big(F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}\Big), \tag{6.11}$$

where the value U_i^n represents the volume averaged quantity over the ith subdomain $[x_{i-1/2}, x_{i+1/2}]$,

$$U_i^n = \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t^n)dx, \tag{6.12}$$

and the flux function $F^n_{i+1/2}$ approximates the time averaged flux along $x_{i+1/2}$,

$$F^n_{i+\frac{1}{2}} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f(u(x_{i+1/2}, t))dt. \tag{6.13}$$

If we sum $\Delta x U_i^{n+1}$ from Eq. 6.11 over the subdomains, we obtain

$$\Delta x \sum_{i=1}^{N} U_i^{n+1} = \Delta x \sum_{i=1}^{N} U_i^n - \Delta t\Big(F^n_{N+\frac{1}{2}} - F^n_{\frac{1}{2}}\Big). \tag{6.14}$$

Here, the sum of the flux differences cancels out except for the two fluxes at the left- and right-most domain boundaries $x = a$ and $x = b$.

Note that the relation in Eq. 6.14 resembles the true conservation law of the exact solution $u(x, t)$ on $[a, b]$ (see Eq. 2.42),

$$\int_a^b u(x, t^{n+1})dx - \int_a^b u(x, t^n)dx = \int_{t^n}^{t^{n+1}} f(u(a, t))dt - \int_{t^n}^{t^{n+1}} f(u(b, t))dt. \tag{6.15}$$

In this way, it makes sense the discrete method in Eq. 6.14 is said to be *in conservation form*. This discrete conservation means that any shocks we compute must, in a sense, be in the "correct" location. Consider, on the other hand, the nonconservative method Eq. 6.4 of Burgers' equation. It is easy to verify that the nonconservative method Eq. 6.4 does not satisfy Eq. 6.14.

Problem 1 Write a simple program for Burgers' equation to implement both the nonconservative and conservative methods in Eq. 6.8 and Eq. 6.9, respectively. Use the two methods to reproduce the results in Fig. 1 and Fig. 2.

2. Consistency for Discontinuous Solutions

In the case of smooth solutions, we made use of Taylor series expansions in order to define consistency. We no longer have a chance to use Taylor series expansions on discontinuous solutions. Instead we need the form of consistency specified as below.

For a hyperbolic problem where information propagates with finite speed, it is convenient to write $F^n_{i\pm1/2}$ as a function of neighboring cell data, e.g., U^n_{i-1}, U^n_i and U^n_{i+1}. See Eq. 4.60 and Eq. 4.61. The formula for conservation law can then be written using the numerical flux of the form

$$F^n_{i+1/2} = \mathcal{F}(U^n_i, U^n_{i+1}). \tag{6.16}$$

Definition: In general (also for discontinuous solutions), the method

$$U^{n+1}_i = U^n_i - \frac{\Delta t}{\Delta x}\Big(\mathcal{F}(U^n_i, U^n_{i+1}) - \mathcal{F}(U^n_{i-1}, U^n_i)\Big) \tag{6.17}$$

is said to be *consistent* with the original conservation law if the numerical flux function $\mathcal{F}$ reduces to the true flux f for the case of constant flow. That is to say, if the exact solution $u(x,t) \equiv \bar{u}$ is constant, then from Eq. 6.13, we expect

$$\mathcal{F}(\bar{u}, \bar{u}) = f(\bar{u}), \forall \bar{u} \in \mathbb{R}. \tag{6.18}$$

This is part of the basic consistency condition.

Remark: We generally also expect continuity in this function as U^n_{i-1}, U^n_i vary, so that

$$\lim_{U^n_{i-1}, U^n_i \to \bar{u}} \mathcal{F}(U^n_{i-1}, U^n_i) = f(\bar{u}). \tag{6.19}$$

For this, we can impose some requirement of *Lipschitz continuity* defined as follow.

Definition: We say $\mathcal{F}$ is Lipschitz at $\bar{u}$ if there exist a constant L (which may depend on $\bar{u}$) such that

$$|\mathcal{F}(U^n_{i-1}, U^n_i) - f(\bar{u})| \le L \max\Big(|U^n_i - \bar{u}|, |U^n_{i-1} - \bar{u}|\Big) \tag{6.20}$$

for all U^n_{i-1} and U^n_i with $|U^n_i - \bar{u}|$ and $|U^n_{i-1} - \bar{u}|$ sufficiently small. We say $\mathcal{F}$ is Lipschitz continuous if it is Lipschitz at every point.

Note: For consistency it suffices to have $\mathcal{F}$ a Lipschitz continuous of each variable.

3. The Lax-Wendroff Theorem

The above discussion suggests that we can hope to correctly approximate discontinuous weak solutions to the conservation law by using a conservative discrete scheme.

Lax and Wendroff proved – the Lax-Wendroff theorem – that this is indeed true, at least in the sense that *if* a discrete solution converges to some function $u(x,t)$ as the grid is refined, then this function $u(x,t)$ will in fact be a weak solution of the conservation law.

It should be noted that the theorem does not guarantee convergence. For that we need some form of stability, and even then, if there is more than one weak solution it might be that one sequence of approximations will converge to one weak solution, while another sequence converges to a different weak solution.

Nonetheless, this is a very powerful and important theorem because it says that we can have confidence in solutions we compute. In practice we compute a single approximation on one fixed grid. If this solution looks reasonable and has well-resolved discontinuities – an indication that the method is stable and our grid is sufficiently fine – then we can believe that it is in fact a good approximation to some weak solution.

We now state the theorem without proof. For complete proof please see our main textbook by LeVeque.

Theorem: (Lax and Wendroff) Consider a sequence of grids indexed by $i, n = 1, 2, ...$, with mesh parameters Δx and Δt vanish to zero as i and n approaches to ∞. Let U_i^n denote the numerical approximation computed with a consistent and conservative method on the ith grid and at nth time step. Suppose that U_i^n converges to a function u as $i, n \to \infty$, in the sense made precise below. Then $u(x, t)$ is a weak solution of the conservation law.

We will assume that we have convergence of U_i^n to u in the following sense:

1. Over every bounded set $\Omega = [a, b] \times [0, T]$ in x-t space,

$$\int_0^T \int_a^b |U_i^n(x,t) - u(x,t)| dx dt \to 0 \text{ as } i, n \to \infty. \tag{6.21}$$

   This is the l^1-norm over the set Ω, so that we can simply write

$$||U_i^n - u||_{1,\Omega} \to 0 \text{ as } i, n \to \infty. \tag{6.22}$$

2. We also assume a property called *Total Variation Bounded, or TVB* that for each T there is an $R > 0$ such that

$$TV(U_i^n(\cdot, t)) < R, \text{ for all } 0 \le t \le T, i, n = 1, 2, \cdots \tag{6.23}$$

   Here TV denotes the total variation function defined as

$$TV(v) = \sup \sum_{j=1}^{N} |v(\xi_j) - v(\xi_{j-1})| \tag{6.24}$$

   where the supremum is taken over all subdivisions of there real line

$$-\infty = \xi_0 < \xi_1 < \cdots < \xi_N = \infty. \tag{6.25}$$

Note: The Lax-Wendroff theorem does not guarantee that weak solutions obtained in this manner satisfy the entropy condition, and there are many examples of conservative numerical methods that converge to weak solutions violating the

entropy condition.

Example: Consider Burgers' equation with initial data

$$u_0(x) = \begin{cases} -1 \text{ for } x < 0, \\ 1 \text{ for } x > 0. \end{cases} \tag{6.26}$$

Let us discretize the initial condition by setting

$$U_i^0 = \begin{cases} -1 \text{ for } x_i \leq 0, \\ 1 \text{ for } x_i > 0. \end{cases} \tag{6.27}$$

We know from **Case II** of the Riemann problem in Chapter 3 that a correct entropy satisfying weak solution includes the rarefaction wave described as in Eq. 2.38. But we also can easily verify that the stationary discontinuity $u(x,t) = u_0(x)$ satisfying Eq. 6.26 is also a weak solution. (Why?)

The stationary weak solution can be readily obtained by conservative methods because the Rankine-Hugoniot condition yields the shock speed $s = 0$ due to $f(-1) = f(1)$ for Burgers' equation. This means that there are very natural conservative methods that converge to this latter solution – the entropy violating weak solution – rather than to the physically correct rarefaction wave.

Note: Notice the sensitivity of this numerical solution to our choice of discrete initial data. If we take a different discretization instead of Eq. 6.26, say,

$$U_i^0 = \begin{cases} -1 \text{ for } x_i < 0, \\ 0 \text{ for } x_i = 0, \\ 1 \text{ for } x_i > 0, \end{cases} \tag{6.28}$$

then it turns out that the upwind method defined by the upwind flux

$$F_{i+1/2}^n = \mathcal{F}(U_i^n, U_{i+1}^n) = \begin{cases} f(U_i^n) \text{ if } s_{i+1/2} \geq 0, \\ \\ f(U_{i+1}^n) \text{ if } s_{i+1/2} < 0, \end{cases} \tag{6.29}$$

gives the proper rarefaction wave solution. Here, $s_{i+1/2}$ is the local shock speed given by the Rankine-Hugoniot condition,

$$s_{i+\frac{1}{2}} = \frac{[f]}{[U]} = \frac{f(U_{i+1}^n) - f(U_i^n)}{U_{i+1}^n - U_i^n}. \tag{6.30}$$

4. Godunov's Method for Finite Volume Methods

The first-order upwind method for the constant-coefficient advection

$$U_i^{n+1} = U_i^n - a\frac{\Delta t}{\Delta x}\left(U_i^n - U_{i-1}^n\right) \tag{6.31}$$

can be considered as a special case of the following approach which was originally proposed by Godunov (1959) as a method for solving the nonlinear Euler

equations of gas dynamics – which we will study in later chapters. We are going to refer this approach to as the *REA algorithm*, for reconstruct-evolve-average.

REA Algorithm:

1. Reconstruct a peicewise polynomial function $\tilde{u}(x, t^n)$ defined for all x, from the cell averages U_i^n. In the simplest case this is a piecewise constant function that takes the value U_i^n in the ith grid cell, i.e.,

$$\tilde{u}(x, t^n) = U_i^n, \forall x \in [x_{i-1/2}, x_{i+1/2}]. \tag{6.32}$$

2. Evolve the hyperbolic equation exactly (or approximately) with this initial data in order to obtain $\tilde{u}(x, t^{n+1})$ a time Δt later.

3. Average this function over each grid cell to obtain new cell averages

$$U_i^{n+1} = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \tilde{u}(x, t^{n+1}) dx. \tag{6.33}$$

The REA algorithm then repeats this process in the next time step.

The Godunov's method is credited with the first successful conservative extension of the Courant-Isaccson-Rees (CIR) scheme (1952) to nonlinear system of conservation laws,

$$U_i^{n+1} = U_i^n - \frac{f'(U_i^n)\Delta t}{\Delta x}\Big(U_i^n - U_{i-1}^n\Big). \tag{6.34}$$

Remark: The CIR scheme was one of the first methods that attempted using upwinding for the equations of gas dynamics, using a linear interpolation based on the two nearest grid values, which are (U_{i-1}^n, U_i^n) and (U_i^n, U_{i+1}^n), depending on whether the corresponding characteristic speed is positive or negative, e.g., $f'(U_i^n) > 0$ or $f'(U_i^n) < 0$.

In other words, if $f'(U_i^n) > 0$ we would interpolate between U_{i-1}^n and U_i^n, giving

$$U_i^{n+1} = \Big[\Big(1 - \frac{f'(U_i^n)\Delta t}{\Delta x}\Big)U_i^n + \frac{f'(U_i^n)\Delta t}{\Delta x} U_{i-1}^n\Big] \tag{6.35}$$

which is another form of Eq. 6.34 as a linear combination. This can be thought as a natural way to achieve upwind approximation to $u_t + f'(u)u_x = 0$, but this is not a good method for a problem with shocks, since it is not in conservation form.

Godunov's first-order upwind method is of the form Eq. 6.11, where the intercell numerical fluxes $F_{i+\frac{1}{2}}^n$ are computed using solutions of *local Riemann problems*. The basic assumption of the method is that at a given time level n the data has a piecewise constant distribution of the form Eq. 6.12, as illustrated in Fig. 3.

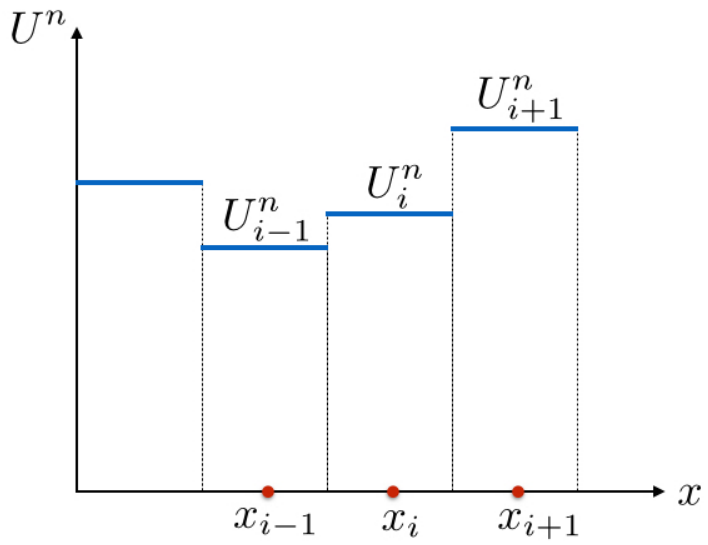


Figure 3. Piecewise constant distribution of data at time $t = t^n$.

The data at time level n may be seen as pairs of constant states (U_i^n, U_{i+1}^n) separated by a discontinuity at the intercell boundary $x_{i+1/2}$. Then locally, one can define a Riemann problem (or a local RP):

$$\text{PDE: } u_t + f(u)_x = 0 \tag{6.36}$$

$$\text{IC: } u(x,0) = u_0(x) = \begin{cases} U_i^n \text{ if } x < x_{i+1/2}, \\ U_{i+1}^n \text{ if } x > x_{i+1/2}. \end{cases} \tag{6.37}$$

This local RP may be solved analytically, if desired. Thus at a given time level n, we have the local RP $RP(U_i^n, U_{i+1}^n)$ with initial data (U_i^n, U_{i+1}^n).

We are now ready for the next step of finding the solution of the *global* problem at the next time level $n+1$, a way of combining the local RPs to produce the global update of U_i^{n+1}.

One way to obtain the global Godunov solution is to write the local RPs in the conservative form, and focus on computing upwind stable Godunov fluxes. Applying the exact same steps for $\tilde{u}(x,t)$ as discussed in considering Eq. 6.11 – Eq. 6.15 – i.e., replace u with $\tilde{u}$ in Eqs. 6.11, 6.12, 6.13, and 6.15, where $\tilde{u}(x,t)$ is understood as the combined solution of $RP(U_{i-1}^n, U_i^n)$ and $RP(U_i^n, U_{i+1}^n)$ and is a true solution satisfying Eq. 6.15 – we arrive to define intercell Godunov fluxes $F_{i\pm1/2}^{n,God}$

$$F_{i-\frac{1}{2}}^{n,God} = \frac{1}{\Delta t}\int_0^{\Delta t} f(\tilde{u}(x_{i-1/2}, t))dt, \tag{6.38}$$

$$F_{i+\frac{1}{2}}^{n,God} = \frac{1}{\Delta t}\int_0^{\Delta t} f(\tilde{u}(x_{i+1/2}, t))dt. \tag{6.39}$$

Note here that we can compute the above integrals *exactly*, since $\tilde{u}(x_{i\pm1/2}, t)$ are constant over the time interval $[0, \Delta t]$. This is because $\tilde{u}(x_{i\pm1/2}, t)$ are the solutions of $RP(U_{i-1}^n, U_i^n)$ and $RP(U_i^n, U_{i+1}^n)$, and each of the RP centered at $x_{i+1/2}$ has a similarity solution that is constant along rays $(x - x_{i+1/2})/t = constant$. And especially, for instance, the value $\tilde{u}(x_{i+1/2}, t)$ is constant along $(x - x_{i+1/2})/t = 0$, which represents each intercell boundary.

If we denote the (constant) solution $\tilde{u}(x_{i+1/2}, t)$ of $RP(U_i^n, U_{i+1}^n)$ by $u_{i+1/2}^*$ at the interfaces, all that requires to construct the Godunov scheme is obtain the intercell Godunov fluxes

$$F_{i\pm\frac{1}{2}}^{n,God} = f\Big(\tilde{u}(x_{i\pm\frac{1}{2}}, t)\Big) = f\Big(u_{i\pm\frac{1}{2}}^*\Big) \tag{6.40}$$

Note that the above Godunov flux is consistent with f because if $U_i^n = U_{i+1}^n \equiv \bar{u}$ then $u_{i\pm\frac{1}{2}}^* = \bar{u}$ as well.

Example: For the constant-coefficient linear scalar conservation law, $u_t + a u_x = 0$ the Godunov flux becomes the standard upwind flux,

$$F_{i+\frac{1}{2}}^{n,God} = f\Big(u_{i+\frac{1}{2}}^*\Big) = f\Big(RP(U_i^n, U_{i+1}^n)\Big) = \begin{cases} f(U_i^n) = aU_i^n & \text{if } a > 0, \\ f(U_{i+1}^n) = aU_{i+1}^n & \text{if } a < 0. \end{cases} \tag{6.41}$$

Note: When applied to the scalar conservation law with $f(u) = au$, Godunov's scheme reduces to the CIR scheme.

Example: Let's consider the Godunov's method for Burgers' equation in the context of nonlinear PDEs. We seek for the solution $u_{i\pm1/2}^*$ of $RP(U_i^n, U_{i+1}^n)$ in two cases, first when $U_i^n \geq U_{i+1}^n$; and second when $U_i^n < U_{i+1}^n$. Recalling the Riemann solutions Eq. 2.35 and Eq. 2.38 from Chapter 3, we get

- $U_i^n \geq U_{i+1}^n$:

$$F_{i+\frac{1}{2}}^{n,God} = f\Big(u_{i+\frac{1}{2}}^*\Big) = \begin{cases} f(U_i^n) & \text{if } s > (x - x_{i+1/2})/t, \\ f(U_{i+1}^n) & \text{if } s < (x - x_{i+1/2})/t, \end{cases} \tag{6.42}$$

  where $s = [f]/[U] = \frac{1}{2}\Big(U_i^n + U_{i+1}^n\Big)$ is the shock speed for Burgers' equation.

- $U_i^n < U_{i+1}^n$:

$$F_{i+\frac{1}{2}}^{n,God} = f\Big(u_{i+\frac{1}{2}}^*\Big) = \begin{cases} f(U_i^n) & \text{if } (x - x_{i+1/2})/t \ \leq U_i^n, \\ f\Big(\frac{x - x_{i+1/2}}{t}\Big) & \text{if } U_i^n < (x - x_{i+1/2})/t \ < U_{i+1}^n, \\ f(U_{i+1}^n) & \text{if } (x - x_{i+1/2})/t \ \geq U_{i+1}^n, \end{cases} \tag{6.43}$$

Note that since we are evaluating Godunov fluxes at $x = x_{i+1/2}$, the above formulas become even much simpler:

- $U_i^n \geq U_{i+1}^n$:

$$F_{i+\frac{1}{2}}^{n,God} = f\Big(u_{i+\frac{1}{2}}^*\Big) = \begin{cases} f(U_i^n) & \text{if } s \geq 0 \\ f(U_{i+1}^n) & \text{if } s < 0. \end{cases} \tag{6.44}$$

- $U_i^n < U_{i+1}^n$:

$$F_{i+\frac{1}{2}}^{n,God} = f\Big(u_{i+\frac{1}{2}}^*\Big) = \begin{cases} f(U_i^n) & \text{if } 0 \ \leq U_i^n, \\ f(0) & \text{if } U_i^n < 0 < U_{i+1}^n, \\ f(U_{i+1}^n) & \text{if } 0 \geq U_{i+1}^n, \end{cases} \tag{6.45}$$

The Godunov's method can be easily understood if we consider all five possible wave patterns in the solution of the Riemann problem for Burgers' equation. These are illustrated in Fig. 4.

If the solution is a shock wave then cases (a) and (c) can occur. The sought value $u_{i+1/2}^*$ on the t-axis depends on the sign of the shock speed s. If the solution is a rarefaction wave then the three possible cases are shown in (b), (d), and (e). Applying terminology from gas dynamics to the rarefaction cases, Figs. (b) and (d) are called *supersonic to the left and the right*, respectively. The case of Fig. (e) is what is called the *transonic rarefaction or sonic rarefaction*; as the wave is crossed, there is a sign change in the characteristic speed u and thus there is one point u_s at which its characteristic speed becomes 0, or $u_s = 0$, a *sonic point*.

Remark: In general, if we assume the flux function $f(u)$ is convex (or concave), i.e., $f''(u)$ does not change sign over the range of u of interest, the sonic point u_s is the (unique) value of u for which $f'(u_s) = 0$. Because of this, u_s is also called the *stagnation point*, since the value u_s propagates with velocity 0.

Remark: Let us remind that u_s is also called the *sonic point*, since in gas dynamics the eigenvalues $u \pm c_s$ where u and c_s are respectively the flow velocity and sound speed, can take value 0 only when the fluid speed $|u|$ is equal to the sound speed c_s.

Remark: The solution shown in Fig. 4 (e) is called a *transonic rarefaction* since in gas dynamics the fluid is accelerated from a subsonic velocity to a supersonic velocity through such a rarefaction. In a transonic rarefaction the value along $(x - x_{i+1/2})/t = 0$ is simply u_s.

Quick summary: What is novel about the Godunov's method is to utilize the local Riemann problems at intercell boundaries in order to compute the global solution of the conservation laws.

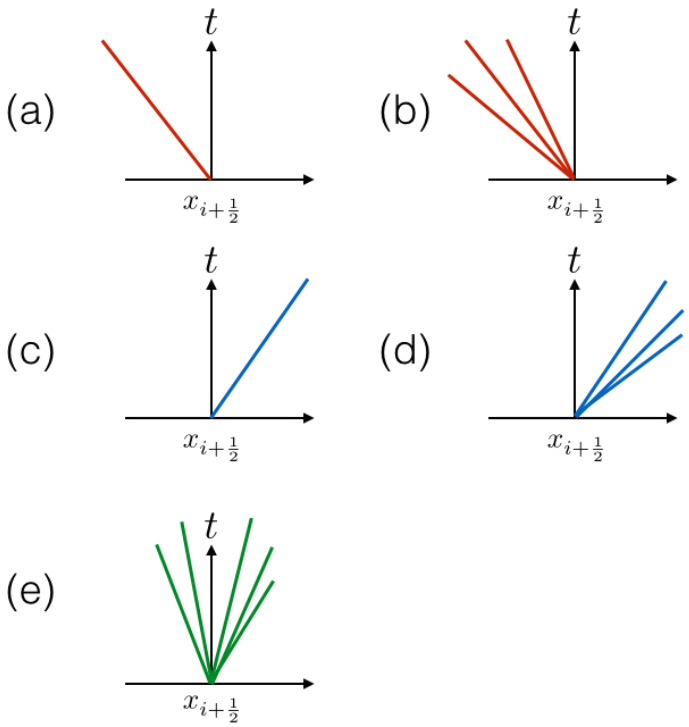


Figure 4. Five possible wave patterns, shown in the x-t plane, in the solution of the Riemann problem for Burgers equation when evaluating the Godunov flux at $x_{i+1/2}$. (a) left-going shock, $u^*_{i+\frac{1}{2}} = U^n_{i+1}$, (b) left-going rarefaction, $u^*_{i+\frac{1}{2}} = U^n_{i+1}$, (c) right-going shock, $u^*_{i+\frac{1}{2}} = U^n_i$, (d) right-going rarefaction, $u^*_{i+\frac{1}{2}} = U^n_i$, and (e) transonic rarefaction, $u^*_{i+\frac{1}{2}} = (x - x_{i+1/2})/t$.

Example: We solve Burgers' equation $u_t + \left(\frac{u^2}{2}\right)_x = 0$ on $[0, 1]$ using the first-order Godunov's method as we described above. The initial condition is given by

$$u(x, 0) = u_0(x) = \begin{cases} u_L = -1 & \text{if } x < x_d, \\ u_R = 2 & \text{if } x \geq x_d, \end{cases} \tag{6.46}$$

where $x_d = 0.5$ is a location of the initial discontinuity. As discussed above the first-order Godunov's method converges this initial condition to a weak solution that includes a rarefaction wave, centered at the initial discontinuity location $x = 0.5$.

Shown in Fig. 5 are the exact rarefaction solution (black curve) and a computed numerical solution on a grid resolution of $N = 32$ using CFL number $C_a = 0.9$. We note that the exact solution can be computed on a given grid

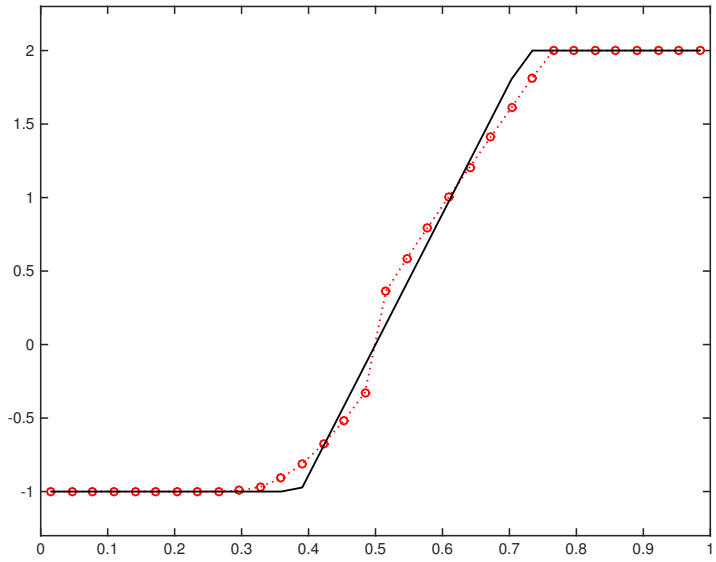


Figure 5. The solid black curve is the entropy-satisfying exact solution to Burgers' equation with a transonic rarefaction wave. The red circle with dotted line show computed solution using the first-order Godunov's method. The results self-similar and are obtained using $t = t_{max} = 0.1$.

resolution as,

$$u_{exact}(x,t) = \begin{cases} u_L & \text{if } x \leq x_L, \\ \frac{x-x_d}{t} & \text{if } x_L < x < x_R, \\ u_R & \text{if } x \geq x_R, \end{cases} \tag{6.47}$$

where $x_L = x_d + u_L t$ and $x_R = x_d + u_R t$ are respectively the left and right most locations that define the region where the time-dependent rarefaction wave is located.

On the other hand, the numerical solution can be obtained by implementing the Godunov flux as described in Eq. 6.44 and Eq. 6.45 using the true flux function $f(u) = u^2/2$ of Burgers equation. Notice that there is a little start-up error existing two closest cells at the initial discontinuity location $x_d = 0.5$ which is due to a discretization error in resolving the stagnation point (or sonic point), at which a vacuum condition is occurred numerically.

Problem 2 Implement the first-order Godunov's method and reproduce the results in the previous example.

Chapter 7

High-Order Methods for Scalar Conservation Laws

The central idea in this chapter is the resolution of two contradictory requirements on numerical methods, namely high-order of accuracy and absence of spurious unphysical oscillations in the vicinity of large gradients.

As seen in Chapter 6, it is well-known that high-order linear (constant coefficient) schemes produce unphysical oscillations near discontinuities. A cure for such spurious oscillations is to introduce the class of monotone methods. However, the statement of Godunov's theorem for linear schemes illustrates – which we will study in this chapter – that monotone methods are at most first-order methods and are therefore of limited use. One way of resolving the contradiction is by constructing *Total Variation Diminishing Methods*, or TVD Methods for short.

In this chapter, we mainly discuss on the scalar conservative PDEs,

$$u_t + f(u)_x = 0. \tag{7.1}$$

For the purpose of this Chapter, we choose $f(u)$ to be the flux for the linear constant advection $f(u) = au$ most of the time. However, nonlinear cases are also generally considered.

We mainly focus on building the class of second-order accurate methods of the Godunov's first-order upwind method. The monotonicity control is realized by introducing the use of *limiter* that changes the magnitude of diffusive behaviors depending on the characteristic structure of the solutions, i.e., smooth or discontinuous. We would like to apply the limiters in such a way that the discontinuous portion of the solution remains non oscillatory, while the smooth portion remains second-order accurate.

It is worth to be noted that the slope-limiter type approach was first pioneered in van Leer's work in his series of five papers, *Towards the ultimate conservative difference scheme I, II, III, IV, V*, over the period of 1973 through

1979.

For the scalar advection equation there are many ways to interpret the same method, in particular, we will see how the slope-limiter approach relates to flux-limiter methods of the type studied by Sweby (1984). The general ideas of these slope-limiter and flux-limiter methods can be extended to the systems of linear and nonlinear equations.

Before proceeding further, let us first define a couple of key concepts related to stability of the (nonlinear) scalar conservation laws.

Definition: A two-level method of the form

$$U_i^{n+1} = \mathcal{N}(U_{i-r+1}^n, \cdots, U_{i+s}^n) \tag{7.2}$$

with nonnegative integers r and s, is said to be a *Total Variation Diminishing (TVD) scheme*, if

$$TV(U^{n+1}) \leq TV(U^n), \forall n. \tag{7.3}$$

Definition: Schemes of the form in Eq. 7.2 is called *monotone* if

$$\frac{\partial \mathcal{N}}{\partial U_k^n} \geq 0, \forall k. \tag{7.4}$$

Example: The Lax-Friedrichs scheme for $u_t + au_x = 0$ can be written as

$$U_i^{n+1} = \frac{1}{2}\Big(1 + C_a\Big)U_{i-1}^n + \frac{1}{2}\Big(1 - C_a\Big)U_{i+1}^n. \tag{7.5}$$

One can easily check LF satisfies the relation in Eq. 7.4 and it is monotone.

Example: The Lax-Wendroff scheme for $u_t + au_x = 0$, written as

$$U_i^{n+1} = \frac{1}{2}C_a\Big(1 + C_a\Big)U_{i-1}^n + \Big(1 - C_a^2\Big)U_i^n - \frac{1}{2}C_a\Big(1 - C_a\Big)U_{i+1}^n. \tag{7.6}$$

is *not monotone*, since it fails to satisfy the relation in Eq. 7.4.

Definition: Schemes of the form in Eq. 7.2 for the scalar, nonlinear conservation law are said to be *Monotonicity Preserving Schemes, or MPS*, provided that

$$U_i^n \geq U_{i+1}^n, \forall i, \tag{7.7}$$

implies that

$$U_i^{n+1} \geq U_{i+1}^{n+1}, \forall i, \tag{7.8}$$

Remark: Any TVD method is MPS.

Remark: A fundamental property of the exact solution of the nonlinear scalar conservation law, when the initial data $u(x, 0)$ has bounded total variation (TVB) (see Eq. 6.23), is

1. no new local extrema in x may be created,

2. the value of a local minimum increases (it does not decrease) and the value of a local maximum decreases (it does not increase).

Remark: TVB can be considered as one of the weakest nonlinear stability condition, in a sense that TVB ensures that a method does not blow up, at least not in an oscillatory manner. TVB does allow large oscillations provided only that spurious oscillations do not grow unboundedly large as time increases.

Remark: For linear methods, if the solution does not blow up then it must either shrink or stay the same size. For nonlinear methods, however, have a much richer variety of behaviors. When you say that a nonlinear method does not blow up, or equivalently, the method is TVB, the method could still have other provocative behaviors, such as, in theory, the error could start small and grow in time, provided the error eventually stopped growing or reached a horizontal asymptote, no matter how large that asymptote might be.

Remark: Let S_{mon} be the set of monotone schemes, S_{tvd} be the set of TVD schemes, S_{tvb} be the set of TVB schemes, and S_{mps} be the set of monotonicity preserving schemes. Then in general, one can show

$$S_{mon} \subseteq S_{tvd} \subseteq S_{tvb} \subseteq S_{mps} \tag{7.9}$$

1. The REA algorithm with Piecewise Linear Reconstruction

Recall the reconstruct-evolve-average (REA) algorithm introduced in the previous chapter. For the scalar advection equation we derived the upwind method by

1. reconstructing a piecewise 'constant' function $\tilde{u}(x, t^n)$ from the cell averages U_i^n,

2. solving the advection equation with $\tilde{u}(x, t^n)$,

3. averaging the result at time t^{n+1} over each grid cell to obtain U_i^{n+1}.

We now work out the three steps in details as follows:

Step 1: Reconstruction
In order to achieve better than first-order accuracy, we must use a better reconstruction, the first step in REA, than a piecewise 'constant' function. That is to say, we now replace the first-order constant reconstruction

$$\tilde{u}(x, t^n) = U_i^n \tag{7.10}$$

with a second-order 'linear' reconstruction

$$\tilde{u}(x, t^n) = U_i^n + \Delta_i^n (x - x_i), x \in I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \tag{7.11}$$

where Δ_i^n is the slope on the ith cell.
Several choices of Δ_i^n include:

$$\Delta_i^n = \begin{cases} \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x} & \text{centered slope,} \\ \frac{U_i^n - U_{i-1}^n}{\Delta x} & \text{upwind slope,} \\ \frac{U_{i+1}^n - U_i^n}{\Delta x} & \text{downwind slope.} \end{cases} \tag{7.12}$$

These are non TVD slope limiters, and thus may generate oscillations near discontinuities.

Those that are TVD slope limiters are available:

$$\Delta_i^n = \begin{cases} \text{minmod}\Big(\frac{U_i^n - U_{i-1}^n}{\Delta x}, \frac{U_{i+1}^n - U_i^n}{\Delta x}\Big) & \text{minmod,} \\ \text{minmod}\Big(\frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}, 2\frac{U_{i+1}^n - U_i^n}{\Delta x}, 2\frac{U_i^n - U_{i-1}^n}{\Delta x}\Big) & \text{MC limiter,} \\ \text{vanLeer}\Big(\frac{U_i^n - U_{i-1}^n}{\Delta x}, \frac{U_{i+1}^n - U_i^n}{\Delta x}\Big) & \text{van Leer's limiter.} \end{cases} \tag{7.13}$$

where the minmod function of two argument is defined by

$$\text{minmod}(a,b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0, \\ b & \text{if } |b| < |a| \text{ and } ab > 0, \\ 0 & \text{if } ab < 0. \end{cases} \tag{7.14}$$

The vanLeer function of two argument is defined as a harmonic mean,

$$\text{vanLeer}(a,b) = \begin{cases} 0 & \text{if } ab \le 0, \\ \frac{2ab}{a+b} & \text{otherwise .} \end{cases} \tag{7.15}$$

In the above, the MC limiter (monotonized central-differencing limiter) gives the sharpest possible slope value in the vicinity of steep gradients; while the minmod limiter the smoothest among three. The van Leer's limiter is in the middle.

Note that the linear reconstruction is defined in such a way that

- the value at cell center is U_i^n: $\tilde{u}(x_i, t^n) = U_i^n$.
- the cell average is U_i^n:

$$\frac{1}{\Delta x}\int_{I_i} \tilde{u}(x, t^n)dx = U_i^n. \tag{7.16}$$

These two facts are crucial in developing conservative methods for conservation laws, because in order that the REA algorithm to be conservative we should use a *conservative reconstruction* in Step 1. And these two facts guarantee that the linear reconstruction *is* conservative indeed. Note that Step 2 and Step 3 are conservative in general.

From Eq. 7.11, we see that

$$\tilde{u}(x_{i+\frac{1}{2}}, t^n) = \begin{cases} U_i^R = U_i^n + \frac{\Delta x}{2}\Delta_i^n & \text{if } a > 0, \\ U_{i+1}^L = U_{i+1}^n - \frac{\Delta x}{2}\Delta_{i+1}^n & \text{if } a < 0. \end{cases} \tag{7.17}$$

Step 2: Evolve
For the scalar advection equation $u_t + au_x = 0$, $a > 0$, we can easily solve the equation for $\tilde{u}(x, t^{n+1})$ with the data $\tilde{u}(x, t^n)$ by tracing back to cell I_i:

$$\tilde{u}(x, t^{n+1}) = \tilde{u}(x - a\Delta t, t^n), \forall x \in I_i. \tag{7.18}$$

Similarly, for $a < 0$, we trace back to cell I_{i+1}:

$$\tilde{u}(x, t^{n+1}) = \tilde{u}(x - a\Delta t, t^n), \forall x \in I_{i+1}. \tag{7.19}$$

Step 3: Average
For the scalar advection equation $u_t + au_x = 0$, $a > 0$, we can easily solve the equation with the data $\tilde{u}(x, t^n)$, for $t \in [t^n, t^{n+1}]$:

$$\begin{aligned} \tilde{u}(x_{i+\frac{1}{2}}, t) &= \tilde{u}(x_{i+\frac{1}{2}} - a(t - t^n), t^n) && (7.20) \\ &= U_i^n + \Delta_i^n\Big(x_{i+\frac{1}{2}} - a(t - t^n) - x_i\Big) && (7.21) \\ &= U_i^n + \Delta_i^n\Big(\frac{\Delta x}{2} - a(t - t^n)\Big). && (7.22) \end{aligned}$$

Similarly, we get for $a < 0$,

$$\begin{aligned} \tilde{u}(x_{i+\frac{1}{2}}, t) &= U_{i+1}^n + \Delta_{i+1}^n\Big(x_{i+\frac{1}{2}} - a(t - t^n) - x_{i+1}\Big) && (7.23) \\ &= U_{i+1}^n - \Delta_{i+1}^n\Big(\frac{\Delta x}{2} + a(t - t^n)\Big). && (7.24) \end{aligned}$$

In summary we obtain:

$$\tilde{u}(x_{i+\frac{1}{2}}, t) = \begin{cases} \tilde{u}^L = U_i^n + \Delta_i^n\Big(\frac{\Delta x}{2} - a(t - t^n)\Big) & \text{if } a > 0, \\ \tilde{u}^R = U_{i+1}^n - \Delta_{i+1}^n\Big(\frac{\Delta x}{2} + a(t - t^n)\Big) & \text{if } a < 0. \end{cases} \tag{7.25}$$

Especially when $t = t^{n+1/2}$, we get:

$$\tilde{u}(x_{i+\frac{1}{2}}, t^{n+1/2}) = \begin{cases} \tilde{u}^L = U_i^n + \frac{\Delta x}{2}\Delta_i^n\Big(1 - C\Big) & \text{if } a > 0, \\ \tilde{u}^R = U_{i+1}^n - \frac{\Delta x}{2}\Delta_{i+1}^n\Big(1 + C\Big) & \text{if } a < 0, \end{cases} \tag{7.26}$$

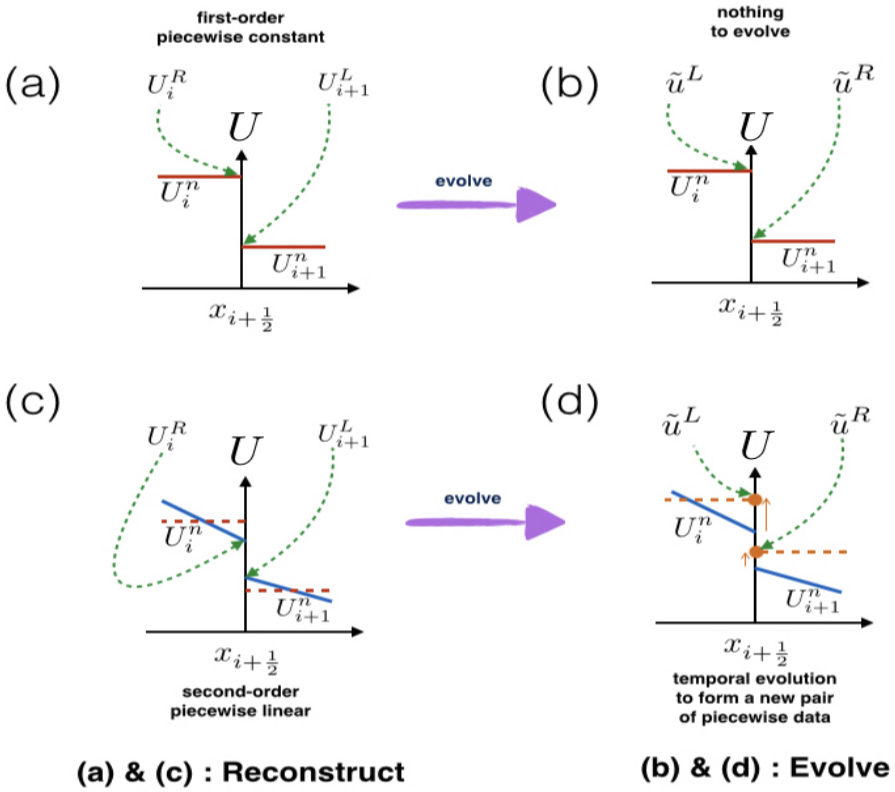


Figure 1. Boundary extrapolated values. First-order update: (a) shows the first-order piecewise constant reconstructed values, $U_i^R = U_i^n$ and $U_{i+1}^L = U_{i+1}^n$ (red lines). (b) In the first-order method, there is nothing to be evolved, and hence a pair of the Riemann states becomes $(\tilde{u}^L, \tilde{u}^R) = (U_i^R, U_{i+1}^L)$. Second-order update: (c) At each interface $x_{i+1/2}$ boundary extrapolated values U_i^R and U_{i+1}^L (blue lines) are reconstructed from the cell averaged values U_i^n and U_{i+1}^n (red dotted lines). Notice that the equal area rule holds, see Eq. 7.16. (d) The reconstructed values U_i^R and U_{i+1}^L are temporally evolved (orange dotted lines and arrows) to a new pair of Riemann states, $(\tilde{u}^L, \tilde{u}^R)$, to form the piecewise constant data for a local Riemann problem at the intercell boundary $x_{i+1/2}$.

where $C = a\Delta t/\Delta x$.

The REA procedure is illustrated in Fig. 1. Using the mid-point quadrature rule in time, we now can give the second-order accurate Godunov-type flux for the piecewise linear method (PLM):

$$F_{i+\frac{1}{2}}^{PLM} = \frac{1}{\Delta t}\int_{t^n}^{t^{n+1}} f\Big(\tilde{u}(x_{i+\frac{1}{2}}, t)\Big)dt = f\Big(\tilde{u}(x_{i+\frac{1}{2}}, \frac{\Delta t}{2})\Big) = \begin{cases} f(\tilde{u}^L) & \text{if } a > 0, \\ f(\tilde{u}^R) & \text{if } a < 0. \end{cases} \tag{7.27}$$

Example: The PLM flux for the linear advection $u_t + au_x = 0$ is

$$F^{PLM}_{i+\frac{1}{2}} = \begin{cases} a\Big[U^n_i + \frac{\Delta x}{2}\Delta^n_i\Big(1 - C\Big)\Big] & \text{if } a > 0, \\ a\Big[U^n_{i+1} - \frac{\Delta x}{2}\Delta^n_{i+1}\Big(1 + C\Big)\Big] & \text{if } a < 0. \end{cases} \tag{7.28}$$

Note: The CFL condition for the linear constant advection equation is simply

$$\Delta t = \frac{C_{\text{cfl}}\Delta x}{|a|}, \tag{7.29}$$

with the constant advection velocity a and a CFL number C_{cfl}. Since a is constant, this calculation does not require any local evaluation on each cell, nor need any temporal update. Therefore Δt can be computed only once at the beginning of simulations.

Remark: The PLM formulation discussed above can be naturally extended to solve Burgers' equation when we replace the constant advection velocity a with the local shock speed $s_{i+1/2} = (u_i + u_{i+1})/2$ at each interface $x_{i+1/2}$ and $C_{i+1/2} = s_{i+1/2}\Delta t/\Delta x$.

Then the PLM Godunov-type flux for Burgers' equation becomes:

- flux for shock solution when $U^n_i \geq U^n_{i+1}$:

$$F^{PLM}_{i+\frac{1}{2}} = \begin{cases} \frac{1}{2}\Big[U^n_i + \frac{\Delta x}{2}\Delta^n_i\Big(1 - C_{i+1/2}\Big)\Big]^2 & \text{if } s_{i+\frac{1}{2}} > 0, \\ \frac{1}{2}\Big[U^n_{i+1} - \frac{\Delta x}{2}\Delta^n_{i+1}\Big(1 + C_{i+1/2}\Big)\Big]^2 & \text{if } s_{i+\frac{1}{2}} < 0. \end{cases} \tag{7.30}$$

- flux for rarefaction solution when $U^n_i < U^n_{i+1}$:

$$F^{PLM}_{i+\frac{1}{2}} = \begin{cases} \frac{1}{2}\Big[U^n_i + \frac{\Delta x}{2}\Delta^n_i\Big(1 - C_{i+1/2}\Big)\Big]^2 & \text{if } 0 \leq U^n_i, \\ 0 & \text{if } U^n_i < 0 < U^n_{i+1}, \\ \frac{1}{2}\Big[U^n_{i+1} - \frac{\Delta x}{2}\Delta^n_{i+1}\Big(1 + C_{i+1/2}\Big)\Big]^2 & \text{if } 0 \geq U^n_{i+1}. \end{cases} \tag{7.31}$$

In this case, the CFL condition should use

$$\lambda_{\max} = \max_i\{|s_{i+\frac{1}{2}}|\} \tag{7.32}$$

to calculate Δt at each time step:

$$\Delta t = \frac{C_{\text{cfl}}\Delta x}{\lambda_{\max}}, \tag{7.33}$$

where C_{cfl} is a CFL number. Unlike the case of the linear constant coefficient advection, Δt now varies both spatially and temporally, hence requiring local propagation speeds $s_{i+1/2}$ and their maximum every time step.

Note: The minmod function can be rewritten in a compact form:

$$\text{minmod}(a,b) = \frac{1}{2}\Big(\text{sign}(a) + \text{sign}(b)\Big)\,\min\Big(|a|,|b|\Big), \tag{7.34}$$

where the sign function is defined by

$$\text{sign}(a) = \begin{cases} 1 & \text{if } a \geq 0, \\ \\ -1 & \text{if } a < 0. \end{cases} \tag{7.35}$$

Note: Likewise, the MC limiter can be rewritten in a compact form:

$$\text{mc}(a,b) = \Big(\text{sign}(a) + \text{sign}(b)\Big)\,\min\Big(\,\min\left(|a|,|b|\right), \frac{1}{4}|a+b|\Big), \tag{7.36}$$

where

$$a = \frac{U_i^n - U_{i-1}^n}{\Delta x}, b = \frac{U_{i+1}^n - U_i^n}{\Delta x}, \tag{7.37}$$

Note: The observation of Eq. 7.28, Eq. 7.30 and Eq. 7.31 gives us that it is straightforward to derive alternative "flux-limiter" formulations from the slope-limiter formulations. In practice we therefore have two options that can be implemented:

1. **Option 1:** More work on calculating the *Riemann states* $\tilde{u}(x_{i+1/2}, t)$ in Eq. 7.25, while keeping the Godunov flux calculation simple as in Eq. 7.27,

2. **Option 2:** Omitting all the Riemann state calculations in Eq. 7.25, and directly implementing the final form of the Godunov fluxes as in Eq. 7.28, Eq. 7.30 and Eq. 7.31.

It seems that two options are exactly equivalent to each other in terms of the needed computational efforts, performance gain, computational conveniency and efficiency. This is true for the current applications of the *scalar* equations, both linear and nonlinear, where the flux functions $f(u)$ are simple in their formulations.

However, for the *system* of equations, both linear and nonlinear, the flux functions $\mathbf{F}(\mathbf{u})$ become a vector quantity that involves more complicated multiple wave structures. Therefore, the flux functions for systems of equations are lot more expensive to calculate numerically.

Many different types of flux formulations have been developed and are available – such as exact Riemann solvers, approximate Riemann solvers, etc. – in

order to provide different levels of computational efficiency, stability, and accuracy. In this sense, if you're considering to explore implementing various types of Riemann solvers, which is the standard practice found in many scientific softwares, it should be much more efficient to keep the flux implementation simpler.

In this way, you provide 'one same' input (i.e., a pair of the left and right Riemann states, $(\tilde{u}^L, \tilde{u}^R)$) to various types of Riemann fluxes at each cell interface $x_{i+1/2}$, rather than implementing 'many different' flux-limiter formulations for many different types of Riemann fluxes.

Example: Nonlinear right-going shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 2 & \text{if } x \leq 0.5, \\ -1 & \text{if } x > 0.5. \end{cases} \tag{7.38}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear standing shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 1 & \text{if } x \leq 0.5, \\ -1 & \text{if } x > 0.5. \end{cases} \tag{7.39}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear left-going shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 1 & \text{if } x \leq 0.5, \\ -2 & \text{if } x > 0.5. \end{cases} \tag{7.40}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear rarefaction
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} -1 & \text{if } x \leq 0.5, \\ 1 & \text{if } x > 0.5. \end{cases} \tag{7.41}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear sine wave evolving to shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \sin(2\pi x) \tag{7.42}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear moving shock and rarefaction
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 2 & \text{if } 0.0 \le x \le 0.3, \\ -1 & \text{if } 0.3 < x \le 0.6, \\ 3 & \text{if } 0.6 < x \le 1. \end{cases} \tag{7.43}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear rarefaction and stationary shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} -1 & \text{if } 0.0 \le x \le 0.3, \\ 2 & \text{if } 0.3 < x \le 0.6, \\ -2 & \text{if } 0.6 < x \le 1. \end{cases} \tag{7.44}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear two right-going shocks evolving into one right-going shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 4 & \text{if } 0.0 \le x \le 0.3, \\ 2 & \text{if } 0.3 < x \le 0.6, \\ -1 & \text{if } 0.6 < x \le 1. \end{cases} \tag{7.45}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear two oppositely-moving shocks evolving into one left-going shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 4 & \text{if } 0.0 \le x \le 0.3, \\ 0 & \text{if } 0.3 < x \le 0.6, \\ -6 & \text{if } 0.6 < x \le 1. \end{cases} \tag{7.46}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Example: Nonlinear two oppositely-moving shocks evolving into one standing shock
Solve Burgers' equation on $[0, 1]$ with IC:

$$u_0(x) = \begin{cases} 4 & \text{if } 0.0 \le x \le 0.3, \\ 0 & \text{if } 0.3 < x \le 0.6, \\ -4 & \text{if } 0.6 < x \le 1. \end{cases} \tag{7.47}$$

Let's see how solutions look like using:
(i) the first-order Godunov method, and
(ii) the second-order PLM method with various slope limiters.

Problem 1 Use the provided conservative first-order Godunov matlab code and extend it to implement the second-order PLM method to solve Burgers' equation with three initial conditions,

(a) a single mode sinusoidal wave over $[0, 1]$:

$$u_0(x) = \sin(2\pi x), \forall x \in [0, 1], \tag{7.48}$$

and
(b) a single shock profile

$$u(x, 0) = u_0(x) = \begin{cases} 2 & \text{if } x < 0.5, \\ -1 & \text{if } x > 0.5, \end{cases} \tag{7.49}$$

(c) a rarefaction wave

$$u(x, 0) = u_0(x) = \begin{cases} -1 & \text{if } x < 0.5, \\ 1 & \text{if } x > 0.5. \end{cases} \tag{7.50}$$

Use grid resolutions of $N = 32, 64, 128$ and compare your results with the results shown in Fig. 1 in Chapter 6. For PLM, you use slope limiters of

(a) all three non TVD limiters (centered, upwind, and downwind) in Eq. 7.12, and
(b) all three TVD limiters (minmod, van Leer's, and MC) in Eq. 7.13.

Please also check that you're getting the following facts when using the non TVD slope limiters:

- the Lax-Wendroff result when using the downwind slope,
- the Beam-Warming result when using the upwind slope,
- the Fromm's result when using the centered slope.

Chapter 8

Finite Volume Methods for the Euler Equations

In this chapter we first study how to solve the *system of linear equations*. The idea is to 'diagonalize' the system of equations, which enables us to decouple the 'system' into a group of separate piece of equations. Expressed in a system of decoupled equations, these equations can be used for us to solve using the knowledges for solving the 'scalar' linear equations we have established so far.

Finally, numerical approach to solve the *system of nonlinear equations* are then achievable by considering local linearizations the nonlinear flux Jacobian matrix.

Our goal in this chapter is to learn numerical methods to solve the Euler equations. Unlike the simple scalar equations (for both linear and nonlinear) the system of nonlinear equations such as the Euler equations, involve more complicated *wave* structures that are multiple. This implies in the system of equations (for both linear and nonlinear) we need to consider much richer characteristic information, which is in contrast with the simple single-wave form in the scalar equation case.

As a result, the Riemann problem itself is comprised of multiple jumps across each characteristic curves. The associated numerical flux calculation, therefore, should account for these jump conditions over the multiple characteristic waves. Consequently, we would require more sophisticated numerical flux formulations in constructing stable Godunov-type fluxes in upwind sense. For this, we acquire to learn two numerical techniques in designing such numerical fluxes: (i) Roe's approximate Riemann solver, and (ii) HLL approximate Riemann solver. As obviously mentioned in their names, these are both *approximate* Riemann solvers.

One can construct yet more involved class of Riemann solvers, so-called the *exact* Riemann solvers. We should understand the meaning of 'exact' not in a way of utilizing analytical form of flux functions, but in a sense of obtaining 'iterative' flux solutions. The topic is beyond the scope of this course and we

are not going to treat this discussion.

On the other hand, there are other various types of 'approximate' Riemann solvers. Several popular examples include fluxes such as other HLL-type of fluxes in the family of HLL-fluxes: HLLC (hydro and MHD), HLLD (MHD only); local Lax-Friedrichs Riemann solver (or also called the Rusanov Riemann solver), two-shock or two-rarefaction Riemann solvers, Osher's numerical flux, a class of Riemann solvers hybridizing more than one type of formulations, and many others.

In general, the choice of selecting *proper* Riemann solvers for real applications significantly affect the outcome of solution accuracy and stability. The numerical behavior of any given Riemann solver may behave differently depending on types of physics problems (e.g., hydro or MHD), dimensionality of problems (e.g., 1D, 2D and 3D), and even on numerical methods themselves (e.g., the first-order Godunov's method, PLM, PPM, numerical viscosity of given methods, dimensionally split vs. unsplit schemes, etc.). For this reason, one needs to choose Riemann solvers carefully before committing real scientific simulations.

Quick summary: Before proceeding further, let's make one quick summary connecting the scalar equations and the system of equations.

- Diagonalizing the system of equations allows us to use the numerical techniques from the scalar equations,
- Linearizing the nonlinear system of equations enables us to use the numerical techniques from the linear system of equations,
- Given multiple characteristic waves in the system, considering each upwind direction in multiple waves makes the numerical solutions of the system stable,
- We basically need to replace the single-wave information (a for the linear advection; $f'(u)$ for Burgers' equation) with the eigenvalues $\lambda_i, i = 1, ..., m$ ($m = 3$ for the Euler equations) of the $m \times m$ flux Jacobian matrix $\mathbf{A} = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}}$.

1. Linear Hyperbolic Systems

In this chapter we begin the study of systems of conservation laws by reviewing the theory of a constant coefficient linear hyperbolic system. Here we can solve the equations explicitly by transforming to characteristic variables – which will be defined later.

Consider the linear system

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0, \tag{8.1}$$
$$\mathbf{U}(x, 0) = \mathbf{U}_0(x), \tag{8.2}$$

where $\mathbf{U} : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^m$ and $\mathbf{A} \in \mathbb{R}^{m \times m}$ is a constant matrix. Notice here we have now vector quantities for the conservative $\mathbf{U}$ and its flux $\mathbf{F}(\mathbf{U}) = \mathbf{A}\mathbf{U}$.

This system of conservation laws is called *hyperbolic* if $\mathbf{A}$ is diagonalizable with real eigenvalues,

$$\lambda_1 \leq \lambda_2 \leq ...\lambda_k \leq \lambda_m, \tag{8.3}$$

so that we can decompose

$$\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1} \tag{8.4}$$

where $\mathbf{\Lambda} = \text{dig}(\lambda_1, ..., \lambda_m)$ is a diagonal matrix of eigenvalues λ_k and $\mathbf{R} = [\mathbf{r}_1|\mathbf{r}_2|...|\mathbf{r}_m]$ is the matrix whose columns are right eigenvectors, $\mathbf{r}_k$.

Note that $\mathbf{A}\mathbf{R} = \mathbf{R}\mathbf{\Lambda}$, i.e.,

$$\mathbf{A}\mathbf{r}_k = \lambda_k \mathbf{r}_k, \text{ for } k = 1, 2, ..., m. \tag{8.5}$$

The system is called *strictly hyperbolic* if the eigenvalues are distinct,

$$\lambda_1 < \lambda_2 < ...\lambda_k < \lambda_m. \tag{8.6}$$

For the most part, especially when considering 1D cases, we make this assumption.

1.1. Diagonalizing the Coupled System into Decoupled System of Linear Equations

Using the diagonalization of Eq. 8.4, we see that the original linear system Eq. 8.1 can be cast into

$$\mathbf{R}^{-1}\mathbf{U}_t + \mathbf{\Lambda}\mathbf{R}^{-1}\mathbf{U}_x = 0, \tag{8.7}$$

where $\mathbf{R}^{-1}$ is constant. Therefore we can further write this as

$$\mathbf{W}_t + \mathbf{\Lambda}\mathbf{W}_x = 0, \tag{8.8}$$

where we define the *characteristic variables* $\mathbf{W}$ as

$$\mathbf{W} = \mathbf{R}^{-1}\mathbf{U}. \tag{8.9}$$

Notice that since $\mathbf{\Lambda}$ is diagonal, this decouples into m *independent scalar equations*

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = 0, k = 1, 2, ..., m, \tag{8.10}$$

where w_k are the components of $\mathbf{W}$

$$\mathbf{W} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \\ \vdots \\ w_m \end{pmatrix}. \tag{8.11}$$

Similarly, we also denote the components of the conserved quantity by

$$\mathbf{U}(x,t) = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \\ \vdots \\ u_m \end{pmatrix}. \tag{8.12}$$

Each of the equations in Eq. 8.10 is a constant coefficient linear advection equation, and this is something we already know how to solve. That is, for each wave k, we obtain its solution

$$w_k(x,t) = w_k(x - \lambda_k t, 0). \tag{8.13}$$

The initial conserved data $\mathbf{U}(x,t)$ is recovered straightforwardly by projecting the characteristic variables $\mathbf{W}(x,t)$ back to the conserved vector space by multiplying $\mathbf{W}$ by $\mathbf{R}$,

$$\mathbf{U}(x,t) = \mathbf{R}\mathbf{W}(x,t) = [\mathbf{r}_1|\mathbf{r}_2|...|\mathbf{r}_m] \begin{pmatrix} w_1(x,t) \\ w_2(x,t) \\ \vdots \\ w_k(x,t) \\ \vdots \\ w_m(x,t) \end{pmatrix}. \tag{8.14}$$

Note from Eq. 8.14 that the value $w_k(x,t)$ is the coefficient of r_k in an eigenvector expansion of the vector $\mathbf{U}(x,t)$, so that we can write out Eq. 8.14 as

$$\mathbf{U}(x,t) = \sum_{k=1}^{m} \mathbf{r}_k w_k(x,t) = \sum_{k=1}^{m} \mathbf{r}_k w_k(x - \lambda_k t, 0). \tag{8.15}$$

Note: We notice that $\mathbf{R}^{-1}$ is the accompanying left eigenvectors, and shall be denoted by

$$\mathbf{L} = \mathbf{R}^{-1} \tag{8.16}$$

whose rows represent individual k-th component of the left eigenvector $\mathbf{l}_k$, satisfying

$$\mathbf{l}_i \cdot \mathbf{r}_j = \delta_{ij}, \tag{8.17}$$

where δ_{ij} is the Kronecker delta function.

2. Linearization of Nonlinear Systems

Now we consider a nonlinear system of conservation laws

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \tag{8.18}$$

where $\mathbf{U} : \mathbb{R} \times \mathbb{R}^+ \to \mathbb{R}^m$ and $\mathbf{F} : \mathbb{R}^m \to \mathbb{R}^m$. This can be rewritten in the *quasilinear form*

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = 0, \tag{8.19}$$

where the flux Jacobian matrix $\mathbf{A}(\mathbf{U}) = \partial\mathbf{F}(\mathbf{U})/\partial\mathbf{U}$ is the $m \times m$ matrix. Again the system is *hyperbolic* if $\mathbf{A}(\mathbf{U})$ is diagonalizable with real eigenvalues for all values of $\mathbf{U}$, at least in some range where the solution is known to lie, and *strictly hyperbolic* if the eigenvalues are distinct for all $\mathbf{U}$.

A full linearization can be available to the quasilinear form Eq. 8.19 if we further linearize the problem about a constant state $\bar{\mathbf{U}} = \mathbf{U}_{avg}$, and hence obtain a constant coefficient linear system, with the Jacobian matrix frozen at $\mathbf{A}(\bar{\mathbf{U}}) = \mathbf{A}(\mathbf{U}_{avg})$,

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U}_{avg})\mathbf{U}_x = 0, \tag{8.20}$$

As can be easily expected, the constant state $\bar{\mathbf{U}} = \mathbf{U}_{avg}$ represents an averaged state between the left and right Riemann states at each interface when solving the local Riemann problems. We refer to solving Eq. 8.20 as solving the nonlinear system via linearization.

Note: The 'nonlinear' behavior of the linearized system would depend on the choice of the mean averaged constant state $\bar{\mathbf{U}} = \mathbf{U}_{avg}$. In general, this averaged state is not uniquely determined, rather allowing infinitely many possible choices of writing such linearization. For many practical purposes, an simple arithmetic averaging between the left and right states can be used in the evaluation, i.e., $\mathbf{U}_{avg} = (\mathbf{U}_L + \mathbf{U}_R)/2$, although one can provide a better averaging scheme, such as the Roe's averaged state. For our purposes in this course, however, we will adopt the simple arithmetic averaging scheme.

3. The Euler Equations

Here in this section we present the one-dimensional time-dependent Euler equations obeying a simple thermodynamics property, an ideal Equation of State (EoS), using three different formulations.

The first form is in the conservative-variable form by writing the Euler equations in the conservative variables,

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}, \tag{8.21}$$

where ρ is density, ρu is momentum where u is particle velocity, and E is total energy per unit volume,

$$\rho E = \rho\Big(\frac{u^2}{2} + e\Big), \tag{8.22}$$

with e the specific internal energy given by a caloric EoS,

$$e = e(\rho, p). \tag{8.23}$$

The second form is in the primitive-variable form using the primitive variables,

$$\mathbf{V} = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}, \tag{8.24}$$

and lastly, the third form is what we already have derived using the characteristic variables, which can be derived either from the conservative variables

$$\mathbf{W} = (\mathbf{R}^c)^{-1}\mathbf{U} = \mathbf{L}^c\mathbf{U} = \begin{pmatrix} \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3 \end{pmatrix}^c \begin{pmatrix} \rho \\ \rho u \\ \rho E \end{pmatrix}, \tag{8.25}$$

or from the primitive variables,

$$\mathbf{W} = (\mathbf{R}^p)^{-1}\mathbf{V} = \mathbf{L}^p\mathbf{V} = \begin{pmatrix} \mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3 \end{pmatrix}^p \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}. \tag{8.26}$$

Here we introduced the two families of eigenvectors as a pair, the first is for the conservative left and right eigenvectors $(\mathbf{R}^c, \mathbf{L}^c)$ that project between the conservative and characteristic variables, while the second is for the primitive left and right eigenvectors $(\mathbf{R}^p, \mathbf{L}^p)$ that project between the primitive and characteristic variables.

3.1. The Conservative-Variable Form of the Euler Equations

In the conservative formulation, we write the Euler equations in a compact conservative form

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0}, \tag{8.27}$$

or

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = \mathbf{0}, \tag{8.28}$$

where the vector of the conservative variables $\mathbf{U}$ is given by Eq. 8.21 and the vector of the conservative fluxes is given by

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{pmatrix}, \tag{8.29}$$

For ideal gases one has the simple expression for Eq. 8.23,

$$e = e(\rho, p) = \frac{p}{(\gamma - 1)\rho}, \tag{8.30}$$

with γ denoting the ratio of specific heats. We also define the sound speed c_s as

$$c_s = \sqrt{\frac{\gamma p}{\rho}}. \tag{8.31}$$

Writing Eq. 8.27 in the quasilinear form Eq. 8.19, we find the coefficient Jacobian matrix $\mathbf{A}(\mathbf{U})$

$$\mathbf{A}(\mathbf{U}) = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \frac{\partial f_1}{\partial u_2} & \frac{\partial f_1}{\partial u_3} \\ \frac{\partial f_2}{\partial u_1} & \frac{\partial f_2}{\partial u_2} & \frac{\partial f_2}{\partial u_3} \\ \frac{\partial f_3}{\partial u_1} & \frac{\partial f_3}{\partial u_2} & \frac{\partial f_3}{\partial u_3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{u^2}{2}(\gamma-3) & (\gamma-3)u & (\gamma-1) \\ \frac{u^3}{2}(\gamma-2) - \frac{c_s^2 u}{\gamma-1} & \frac{u^2(3-2\gamma)}{2} + \frac{c_s^2}{\gamma-1} & \gamma u \end{bmatrix}. \tag{8.32}$$

We proceed to diagonalize the matrix $\mathbf{A}(\mathbf{U})$ to get

$$\mathbf{L}^c \mathbf{A} \mathbf{R}^c = \mathbf{\Lambda} \tag{8.33}$$

which allows us to write in a decoupled system of equations in terms of the characteristic variables as in Eq. 8.10, or in a compact vector form,

$$\mathbf{W}_t + \mathbf{\Lambda} \mathbf{W}_x = \mathbf{0}. \tag{8.34}$$

It is important to explicitly write out the conservative left and right eigenvectors $(\mathbf{R}^c, \mathbf{L}^c)$ of $\mathbf{A}$ that correspond to the eigenvalues

$$\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3) = (u - c_s, u, u + c_s). \tag{8.35}$$

The right eigenvectors can be found out to be the columns of the right eigenvector matrix

$$\mathbf{R}^c = \begin{bmatrix} -\frac{\rho}{2c_s} & 1 & \frac{\rho}{2c_s} \\ -\frac{\rho}{2c_s}(u - c_s) & u & \frac{\rho}{2c_s}(u + c_s) \\ -\frac{\rho}{2c_s}\left(\frac{u^2}{2} + \frac{c_s^2}{\gamma-1} - c_s u\right) & \frac{u^2}{2} & \frac{\rho}{2c_s}\left(\frac{u^2}{2} + \frac{c_s^2}{\gamma-1} + c_s u\right) \end{bmatrix}, \tag{8.36}$$

and the left eigenvectors are the rows of the left eigenvector matrix

$$\mathbf{L}^c = \frac{\gamma - 1}{\rho c_s} \begin{bmatrix} -\frac{u^2}{2} - \frac{c_s u}{\gamma-1} & u + \frac{c_s}{\gamma-1} & -1 \\ \frac{\rho}{c_s}\left(-\frac{u^2}{2} + \frac{c_s^2}{\gamma-1}\right) & \frac{\rho u}{c_s} & -\frac{\rho}{c_s} \\ \frac{u^2}{2} - \frac{c_s u}{\gamma-1} & -u + \frac{c_s}{\gamma-1} & 1 \end{bmatrix}. \tag{8.37}$$

3.2. The Primitive-Variable Form of the Euler Equations

The primitive formulation of the Euler equations is given as in terms of the primitive variables $\mathbf{V}$ in Eq. 8.24,

$$\mathbf{V}_t + \mathbf{A}(\mathbf{V}) \mathbf{V}_x = \mathbf{0}, \tag{8.38}$$

where we find the coefficient matrix is written as

$$\mathbf{A}(\mathbf{V}) = \begin{bmatrix} \frac{\partial f_1}{\partial v_1} & \frac{\partial f_1}{\partial v_2} & \frac{\partial f_1}{\partial v_3} \\ \frac{\partial f_2}{\partial v_1} & \frac{\partial f_2}{\partial v_2} & \frac{\partial f_2}{\partial v_3} \\ \frac{\partial f_3}{\partial v_1} & \frac{\partial f_3}{\partial v_2} & \frac{\partial f_3}{\partial v_3} \end{bmatrix} = \begin{bmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho c_s^2 & u \end{bmatrix}. \tag{8.39}$$

As in the case of the conservative formulation, we can achieve a decoupled system in terms of the characteristic variables by diagonalizing the matrix $\mathbf{A}(\mathbf{V})$,

$$\mathbf{L}^p \mathbf{A} \mathbf{R}^p = \mathbf{\Lambda}, \tag{8.40}$$

where the right eigenvectors are given by

$$\mathbf{R}^p = \begin{bmatrix} -\frac{\rho}{2c_s} & 1 & \frac{\rho}{2c_s} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{\rho c_s}{2} & 0 & \frac{\rho c_s}{2} \end{bmatrix}, \tag{8.41}$$

and the left eigenvectors are given by

$$\mathbf{L}^p = \begin{bmatrix} 0 & 1 & -\frac{1}{\rho c_s} \\ 1 & 0 & -\frac{1}{c_s^2} \\ 0 & 1 & \frac{1}{\rho c_s} \end{bmatrix}. \tag{8.42}$$

Using these eigenvectors, we can convert the primitive form in Eq. 8.38 to the the exact same diagonalized form in Eq. 8.34 written in the characteristic variables $\mathbf{W}$.

Note that there is a simple relationship between $\mathbf{A}(\mathbf{U})$ and $\mathbf{A}(\mathbf{V})$. First notice that

$$d\mathbf{U} = \mathbf{Q} d\mathbf{V}, \tag{8.43}$$

where

$$\mathbf{Q} = \frac{d\mathbf{U}}{d\mathbf{V}} = \begin{bmatrix} 1 & 0 & 0 \\ u & \rho & 0 \\ \frac{u^2}{2} & \rho u & \frac{1}{\gamma - 1} \end{bmatrix}. \tag{8.44}$$

Similarly,

$$d\mathbf{V} = \mathbf{Q}^{-1} d\mathbf{U}, \tag{8.45}$$

where

$$\mathbf{Q}^{-1} = \frac{d\mathbf{V}}{d\mathbf{U}} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 \\ \frac{1}{2}(\gamma - 1)u^2 & -(\gamma - 1)u & \gamma - 1 \end{bmatrix}. \tag{8.46}$$

Then by chain rule, Eq. 8.28 can be written as

$$\mathbf{Q}\mathbf{V}_t + \mathbf{A}(\mathbf{U})\mathbf{Q}\mathbf{V}_x = \mathbf{0}, \tag{8.47}$$

or

$$\mathbf{V}_t + \mathbf{Q}^{-1}\mathbf{A}(\mathbf{U})\mathbf{Q}\mathbf{V}_x = \mathbf{0}. \tag{8.48}$$

Comparing with Eq. 8.38 we see that

$$\mathbf{A}(\mathbf{V}) = \mathbf{Q}^{-1}\mathbf{A}(\mathbf{U})\mathbf{Q}. \tag{8.49}$$

In other words, the two matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{A}(\mathbf{V})$ are *similar* matrices, whereby they both have the same eigenvalues.

3.3. The Characteristic-Variable Form of the Euler Equations

As seen already, we call the form in Eq. 8.34 the *canonical form* or *characteristic form* of system of the Euler equations. When expressed in terms of $\mathbf{W}$, the original coupled linear system Eq. 8.1 becomes *completely decoupled* into a family of individual scalar equations, Eq. 8.10, where we only need to seek for the *single unknown* $w_k(x,t)$,

$$\frac{\partial w_k}{\partial t} + \lambda_k \frac{\partial w_k}{\partial x} = 0, k = 1, 2, ..., m, \tag{8.50}$$

solving the system is therefore identical to solving the linear advection equation we have studied so far.

4. Riemann Problems for the Linearized Euler Equations

We study the Riemann problem for the conservative, hyperbolic, constant coefficient system (or conveniently assuming the linearized version of nonlinear systems) of the form

$$\mathbf{U}_t + \mathbf{A}\mathbf{U}_x = 0, \tag{8.51}$$

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U}_L & \text{if } x < 0, \\ \mathbf{U}_R & \text{if } x > 0. \end{cases} \tag{8.52}$$

For the sake of clear exposition, we also assume that the system is strictly hyperbolic with the real and distinct eigenvalues,

$$\lambda_1 < \lambda_2 < \cdots \lambda_m. \tag{8.53}$$

4.1. Jump Discontinuity Over Multiple Waves

The structure of the solution of the Riemann problem Eq. 8.51 in the x-t plane is illustrated in Fig. 1 for the Euler equations with $m = 3$. It consists of m waves emanating from the origin, or in general, from each cell-interface where we consider the local Riemann problem, one for each eigenvalue λ_k.

Each k-th wave carries a jump discontinuity in $\mathbf{U}$ propagating with speed λ_k. The primary task is to find the solution in the Riemann fan region – a region surrounded by the left $\mathbf{U}_L$ and right $\mathbf{U}_R$ initial states – which is depicted as two different star states consisting of the single-star state $\mathbf{U}^*$ and the double-star state $\mathbf{U}^{**}$. These two states are naturally formed due to the presence of the triple wave family in the system of the Euler equations. In general when there are more number of waves in the system, for instance $m = 7$ in the system of 1D MHD equations, we see there are 6 Riemann state regions in the Riemann fan.

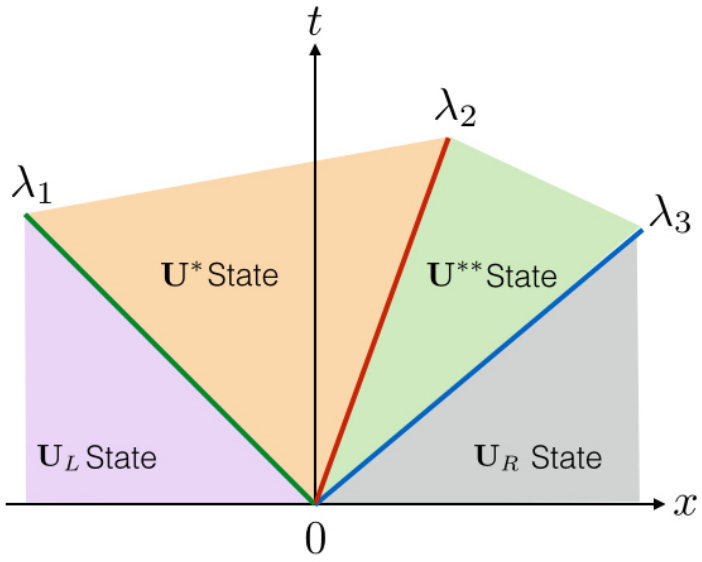


Figure 1. Four different *constant* Riemann state regions of the Riemann fan in solving the Euler equations. Given the left $\mathbf{U}_L$ and right $\mathbf{U}_R$, there are two new star states $\mathbf{U}^*$ and $\mathbf{U}^{**}$ formed from the three characteristic waves (i.e., eigenvalues) of the Euler equations, λ_1, λ_2 and λ_3. The key idea is thus to construct numerical solutions in these two start states from the given left and right initial states by considering each jump discontinuity propagating with speed λ_k.

As the right eigenvectors $\mathbf{r}_k$, $k = 1, 2, 3$, are linearly independent we can expand the left and the right constant states as

$$\mathbf{U}_L = \sum_{i=1}^{3} \alpha_i \mathbf{r}_i \text{ and } \mathbf{U}_R = \sum_{i=1}^{3} \beta_i \mathbf{r}_i, \tag{8.54}$$

with constant coefficients α_i and β_i for $i = 1, 2, 3$.

As considered in the previous section, we conveniently consider the solution of Eq. 8.51 in terms of the characteristic variables w_k and the associated right eigenvectors $\mathbf{r}_k$. We already saw that this allows us to seek for the decoupled

system as in Eq. 8.10. Furthermore, by using the characteristic tracing in Eq. 8.13, followed by the right eigenvector projection, we obtain the result in Eq. 8.15,

$$\mathbf{U}(x,t) = \sum_{k=1}^{m} \mathbf{r}_k w_k(x,t) = \sum_{k=1}^{m} \mathbf{r}_k w_k(x - \lambda_k t, 0). \tag{8.55}$$

Comparing this with the linear independent expansions of the left and the right states Eq. 8.54, we easily see the relation between α_k, β_k and w_k,

$$w_k(x,t) = w_k(x - \lambda_k t, 0) = \begin{cases} \alpha_k & \text{if } x - \lambda_k t < 0, \\ \beta_k & \text{if } x - \lambda_k t > 0. \end{cases} \tag{8.56}$$

We therefore get the final solution to the Riemann problem Eq. 8.51 in the star states in terms of the initial left and right states as

$$\mathbf{U}(x,t) = \sum_{k=1}^{I} \beta_k \mathbf{r}_k + \sum_{k=I+1}^{3} \alpha_k \mathbf{r}_k, \tag{8.57}$$

where the integer I is the maximum value of the sub-index k for which $x - \lambda_k t > 0$, for $\forall k \leq I$.

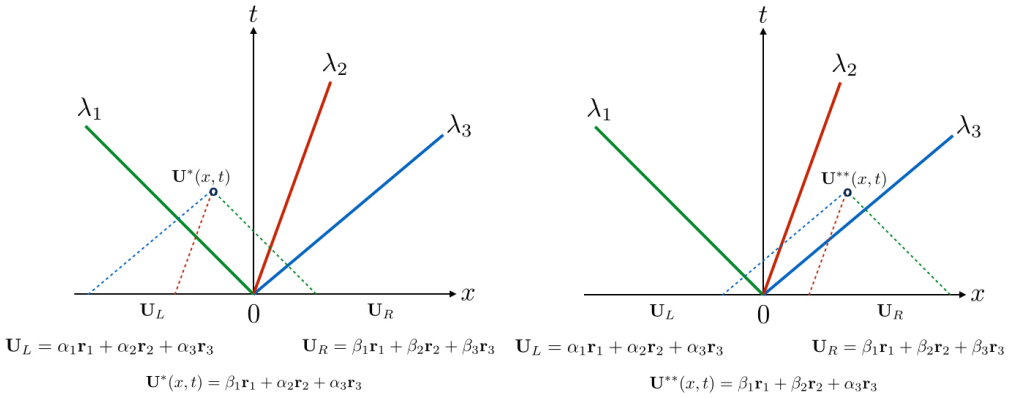


Figure 2. Construction of solution to Riemann problem at the two different star states. Top: The single-star state solution $\mathbf{U}^*$ is obtained by crossing the first wave λ_1 from the left state $\mathbf{U}_L$, giving $\mathbf{U}^* = \beta_1\mathbf{r}_1 + \alpha_2\mathbf{r}_2 + \alpha_3\mathbf{r}_3$. Bottom: The double-star state solution $\mathbf{U}^{**}$ is obtained by crossing the first and the second waves, λ_1 and λ_2, from the left state $\mathbf{U}_L$, giving $\mathbf{U}^{**} = \beta_1\mathbf{r}_1 + \beta_2\mathbf{r}_2 + \alpha_3\mathbf{r}_3$. Alternatively, $\mathbf{U}^{**}$ can also be obtained by crossing the last wave λ_3 from the right state $\mathbf{U}_R$.

As an example, we show in Fig. 2 the Riemann problem solutions at the star states, $\mathbf{U}^*$ and $\mathbf{U}^{**}$. In the top panel of Fig. 2, we look at the solution in the $*$-Region by accounting for a jump discontinuity across the first wave λ_1. Considering the domain of dependence at the point of the location $\mathbf{U}^*$, we see

that there is only one characteristic information (i.e., β_1) emanated from the right initial state $\mathbf{U}_R$ along the characteristic curve $x - \lambda_1 t$ (dotted green line). The other two information (i.e., α_2, α_3) come from the left initial data $\mathbf{U}_L$ along the curves $x - \lambda_2 t$ (dotted red line) and $x - \lambda_3 t$ (dotted blue line). Thus the solution in the $*$-Region, between the λ_1 and λ_2 waves, is

$$\mathbf{U}^* = \beta_1 \mathbf{r}_1 + \alpha_2 \mathbf{r}_2 + \alpha_3 \mathbf{r}_3. \tag{8.58}$$

Similarly, we easily see that the solution in the $**$-Region, between the λ_2 and λ_3 waves, is

$$\mathbf{U}^{**} = \beta_1 \mathbf{r}_1 + \beta_2 \mathbf{r}_2 + \alpha_3 \mathbf{r}_3. \tag{8.59}$$

Now let us consider the total amount of jump $\Delta\mathbf{U}$ in $\mathbf{U}$ across the whole wave structure in the solution of the Riemann problem. It is easy to see that

$$\Delta\mathbf{U} = \mathbf{U}_R - \mathbf{U}_L = \sum_{i=1}^{3} \Big(\beta_i - \alpha_i\Big)\mathbf{r}_i. \tag{8.60}$$

The physical interpretation of this is that the total jump is the sum of individual jumps across the k-th waves, denoted by $\Delta\mathbf{U}_k = (\beta_k - \alpha_k)\mathbf{r}_k$, with $\beta_k - \alpha_k$ the strength of the k-th wave.

4.2. Characteristic Fields

The characteristic speed (or the eigenvalues) $\lambda_k = \lambda_k(\mathbf{U})$ defines a *characteristic field*, the λ_k-field. Sometimes one also speaks of the $\mathbf{r}_k$-field, that is the characteristic field defined the right eigenvector $\mathbf{r}_k$.

There are two different types of characteristic fields:

Definition: A λ_k-characteristic field is said to be *linearly degenerate* if

$$\nabla\lambda_k(\mathbf{U}) \cdot \mathbf{r}_k(\mathbf{U}) = 0, \forall \mathbf{U} \in \mathbb{R}^m. \tag{8.61}$$

Definition: A λ_k-characteristic field is said to be *genuinely nonlinear* if

$$\nabla\lambda_k(\mathbf{U}) \cdot \mathbf{r}_k(\mathbf{U}) \neq 0, \forall \mathbf{U} \in \mathbb{R}^m. \tag{8.62}$$

Note: The gradient of the eigenvalues are simply

$$\nabla\lambda_k(\mathbf{U}) = \nabla_{\mathbf{U}}\lambda_k(\mathbf{U}) = \Big(\frac{\partial\lambda_k}{\partial u_1}, \frac{\partial\lambda_k}{\partial u_2}, \frac{\partial\lambda_k}{\partial u_3}\Big)^T, \tag{8.63}$$

for each k.

Example: The λ_2-characteristic field of the Euler equations is linearly degenerate, since we have

$$\nabla\lambda_2(\mathbf{U}) = \Big(\frac{\partial\lambda_2}{\partial u_1}, \frac{\partial\lambda_2}{\partial u_2}, \frac{\partial\lambda_2}{\partial u_3}\Big) = \Big(-\frac{u}{\rho}, \frac{1}{\rho}, 0\Big), (\text{why???}) \tag{8.64}$$

and

$$\mathbf{r}_2 = \left(1, u, \frac{u^2}{2}\right)^T \tag{8.65}$$

and hence

$$\nabla\lambda_2 \cdot \mathbf{r}_2 = 0. \tag{8.66}$$

Example: In the same manner, one can show that the λ_1- and λ_3-characteristic fields of the Euler equations are both genuinely nonlinearly.

To help our understanding of the two characteristic families, for the moment, let us consider a special case of a solution called a *simple wave* to the conservation law

$$\mathbf{U}(x,t) = \bar{\mathbf{U}}(\xi), \tag{8.67}$$

where $\bar{\mathbf{U}}(\xi)$ is an integral curve with $\xi = \xi(x,t)$. We now define an integral curve.

Definition: We say a smooth curve $\bar{\mathbf{U}}(\xi)$ through state space parametrized by a scalar parameter ξ an *integral curve of the vector field* $\mathbf{r}_k$ if at each point $\bar{\mathbf{U}}(\xi)$ the tangent vector to the curve, $\bar{\mathbf{U}}'(\xi)$, is an eigenvector of $\mathbf{A} = \partial\mathbf{F}/\partial\mathbf{U}$ corresponding to the eigenvalue $\lambda_k(\bar{\mathbf{U}}(\xi))$. Thus we can write $\bar{\mathbf{U}}'(\xi)$ as some scalar multiple of the particular eigenvector $\mathbf{r}_k(\bar{\mathbf{U}}(\xi))$,

$$\bar{\mathbf{U}}'(\xi) = \alpha(\xi)\mathbf{r}_k(\bar{\mathbf{U}}(\xi)). \tag{8.68}$$

The value of $\alpha_k(\xi)$ depends on the particular parameterization of the curve and on the normalization of $\mathbf{r}_k$. Note that the crucial idea is that the tangent to the curve is always in the direction of the appropriate eigenvector $\mathbf{r}_k$ evaluated at the point on the curve.

Example: In Fig. 3 we show a specific case of an illustration of integral curves in the isothermal equations of gas dynamics. This can be considered as a reduced version of the system of the Euler equations with the absence of the energy equation, as the isothermal system does not allow to have evolution of temperature, nor the energy in the system. Then the system becomes, letting $m = \rho u$,

$$\rho_t + m_x = 0 \tag{8.69}$$

$$m_t + \left(\frac{m^2}{\rho} + c_s^2\rho\right)_x = 0. \tag{8.70}$$

We can easily find the flux Jacobian matrix is

$$\frac{\partial\mathbf{F}}{\partial\mathbf{U}} = \begin{bmatrix} 0 & 1 \\ c_s^2 - \frac{m^2}{\rho^2} & \frac{2m}{\rho} \end{bmatrix}, \tag{8.71}$$

with eigenvalues

$$\lambda_1 = \frac{m}{\rho} - c_s, \lambda_2 = \frac{m}{\rho} + c_s, \tag{8.72}$$

and the right eigenvectors

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ \frac{m}{\rho} - c_s \end{pmatrix}, \mathbf{r}_2 = \begin{pmatrix} 1 \\ \frac{m}{\rho} + c_s \end{pmatrix}. \tag{8.73}$$

For convenience, we take $c_s = 1$. The little arrows (blue) indicate a selection of right eigenvectors $\mathbf{r}_1$ with different values of m/ρ that are constant (rays in red). Notice that the eigenvectors are also constant for each of our choices drawn in Fig. 3. The integral curves (dotted green curves) can be drawn by tracing the eigenvectors to which the curves are tangent in the phase plane.

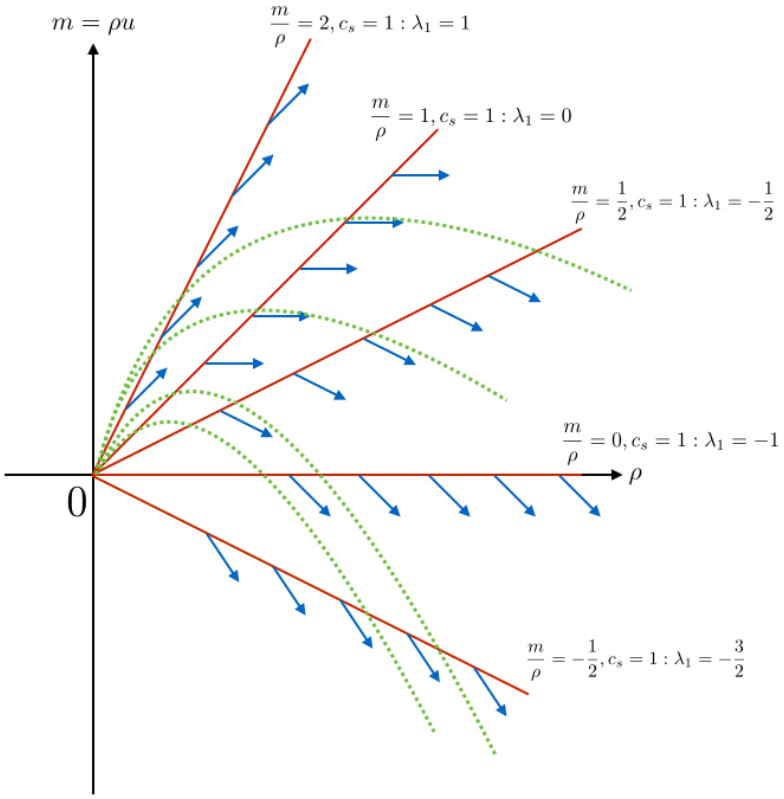


Figure 3. Integral curves of $\mathbf{r}_1$ in the phase plane $(u_1, u_2) = (\rho, \rho u)$. The example illustrates a case for the first characteristic field, $\lambda_1 = u - c_s$, where for exposition purpose we take the sound speed $c_s = 1$.

Now if we evaluate the variation of λ_k along the integral curve Eq. 8.67 we get

$$\frac{d}{d\xi}\lambda_k\left(\bar{\mathbf{U}}(\xi)\right) = \nabla\lambda_k\left(\bar{\mathbf{U}}(\xi)\right) \cdot \bar{\mathbf{U}}'(\xi) \tag{8.74}$$

$$= \nabla\lambda_k\left(\bar{\mathbf{U}}(\xi)\right) \cdot \alpha(\xi)\mathbf{r}_k\left(\bar{\mathbf{U}}(\xi)\right). \tag{8.75}$$

Therefore, that the λ_k-characteristic is linearly degenerate implies that λ_k is identically constant along each integral curve.

Example: Consider a constant-coefficient linear hyperbolic system, in which λ_k is constant everywhere and thus the gradient $\nabla\lambda_k(\mathbf{U}) = 0$ for all $\mathbf{U}$.

Example: Note that for a scalar nonlinear problem $u_t+(f(u))_x = 0$ where there is only one single eigen-structure is available (i.e., $m = 1$), we have $\lambda_1(u) = f'(u)$ and thus can take $\mathbf{r}_1 \equiv 1$. We then see that the definition of genuinely nonlinear field reduces to the convexity requirement

$$f''(u) \neq 0. \tag{8.76}$$

Remark: The wave associated with the λ_2-characteristic field is a *contact discontinuity* and those associated with λ_1- and λ_3-characteristic fields will either be rarefaction waves (smooth) or shock waves (discontinuities). One does not know in advance what types of waves will be present in the solution of the Riemann problem, except for the middle wave, the λ_2-characteristic field which is always a contact discontinuity.

4.3. Elementary-Wave Solutions of the Riemann Problem

For nonlinear systems the waves may be discontinuities such as shock waves and contact waves, or smooth transition waves such as rarefactions. In this section, we simplify our interest and only consider an elementary type of wave solutions which consist of a pair of initial data states $\mathbf{U}_L$ and $\mathbf{U}_R$ connected by a *single* wave.

That is, the solution of the Riemann problem consists of only a single nontrivial wave; rather than involves more than one wave structures (i.e., both shock and rarefaction, or both shock and contact discontinuity, or all of the three). In this way if the wave is a discontinuity we know that the wave must be either a shock or a contact wave. On the other hand, if the wave is smooth, it must be a rarefaction. We now ready to classify these three types of elementary solutions of the Riemann problem in solving the system of equations.

- **Shock Wave:**
  For a shock wave the two constant states $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a single jump discontinuity in a *genuinely nonlinear field k* and the following conditions apply:

  1. the Rankine-Hugoniot Conditions: $\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = s(\mathbf{U}_R - \mathbf{U}_L)$.
  2. the entropy condition: $\lambda_k(\mathbf{U}_L) > s > \lambda_k(\mathbf{U}_R)$

- **Contact Wave:**
  For a contact wave the two constant states $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a single jump discontinuity of speed s in a *linearly degenerate field k* and the following conditions apply:

  1. the Rankine-Hugoniot Conditions: $\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = s(\mathbf{U}_R - \mathbf{U}_L)$.

2. the constancy relation across the wave, called the *Generalized Riemann Invariants across the wave*

$$\frac{dw_1}{\mathbf{r}_k \cdot \mathbf{e_1}} = \frac{dw_2}{\mathbf{r}_k \cdot \mathbf{e_2}} = \frac{dw_3}{\mathbf{r}_k \cdot \mathbf{e_3}}, \tag{8.77}$$

where w_i is the individual component of the (either conservative or primitive) variable $\mathbf{W} = (w_1, w_2, w_3)^T$, $\mathbf{r}_k$ is the k-th right eigenvector (e.g, k-th column of $\mathbf{R}^c$ in Eq. 8.36) corresponding to either conservative or primitive, and $\mathbf{e}_i$ is a unit vector with 1 on its i-th entry while all others are zero (e.g., $\mathbf{e}_3 = (0, 0, 1)$).

3. the parallel characteristic condition $\lambda_k(\mathbf{U}_L) = s = \lambda_k(\mathbf{U}_R)$.

- **Rarefaction Wave:**
  For a rarefaction wave the two constant states $\mathbf{U}_L$ and $\mathbf{U}_R$ are connected through a smooth transition in a *genuinely nonlinear field k* and the following conditions are met:

- the constancy of the Generalized Riemann Invariants across the wave

$$\frac{dw_1}{\mathbf{r}_k \cdot \mathbf{e_1}} = \frac{dw_2}{\mathbf{r}_k \cdot \mathbf{e_2}} = \frac{dw_3}{\mathbf{r}_k \cdot \mathbf{e_3}}. \tag{8.78}$$

- the difference of characteristics $\lambda_k(\mathbf{U}_L) < \lambda_k(\mathbf{U}_R)$.

4.4. Approximate Riemann Solvers for the Euler Equations

For the purpose of designing numerical methods that are conservative when solving systems of the Euler equations we can present two different approaches in terms of implementing Riemann solvers. The first approach is to choose Riemann solvers that evaluate the Riemann solutions in the *exact* way. The resulting Riemann solvers are hence said to be the *exact Riemann solvers*. This way requires to construct the Riemann solutions in the Riemann fan regions by means of considering the elementary-wave solutions case-by-case as and constructing them in the Riemann fan. In practice, these analytical relations often take the form of implicit equations, thus require iterative approaches in computing their exact solutions. This becomes computationally expensive.

On the other hand, as with most practical purposes, one can simplify the Riemann problem by considering an alternative approach using *approximate Riemann solvers* which do not seek for iterative solutions, hence computationally more efficient. The solution to an approximate Riemann solves prove almost as good as or even, in some ways, better (i.e., more numerical stability in most challenging real applications) than the solution to the true Riemann problem, often at a fraction of the cost.

This section we describe two approximate Riemann solvers, the HLL solver and the Roe solver, that replace the true nonlinear flux function by a locally linearized approximation.

4.4.1. HLL Approximate Riemann Solver In order to compute a Godunov flux, Harten, Lax and van Leer presented a novel approach for solving the Riemann problem approximately. The Resulting Riemann solvers have become known as HLL Riemann solvers. In this approach an approximation for the intercell numerical flux is obtained directly, unlike the exact Riemann solvers.

Consider Fig. 4, in which the whole of the wave structure arising from the exact solution of the Riemann problem is contained in the control volume $[x_L, x_R] \times [0, T]$ on x-t plane. The two fastest signal velocities are denoted as S_L and S_R perturbing the initial states $\mathbf{U}_L$ and $\mathbf{U}_R$, respectively. The time T is arbitrarily chosen. The integral form of the conservation laws in Eq. 8.27 in the control volume $[x_L, x_R] \times [0, T]$ becomes

$$\begin{aligned}\int_{x_L}^{x_R} \mathbf{U}(x,T)dx &= \int_{x_L}^{x_R} \mathbf{U}(x,0)dx + \int_0^T \mathbf{F}(\mathbf{U}(x_L,t))dt - \int_0^T \mathbf{F}(\mathbf{U}(x_R,t))dt \\ &= x_R\mathbf{U}_R - x_L\mathbf{U}_L + T\Big(\mathbf{F}_L - \mathbf{F}_R\Big), \end{aligned} \tag{8.79}$$

where $\mathbf{F}_L = \mathbf{F}(\mathbf{U}_L)$ and $\mathbf{F}_R = \mathbf{F}(\mathbf{U}_R)$. We call the integral relation Eq. 8.79 the *Consistency Condition*.

Now splitting the left hand side of Eq. 8.79 into three different integrals,

$$\begin{aligned}\int_{x_L}^{x_R} \mathbf{U}(x,T)dx &= \int_{x_L}^{TS_L} \mathbf{U}(x,T)dx + \int_{TS_L}^{TS_R} \mathbf{U}(x,T)dx + \int_{TS_R}^{x_R} \mathbf{U}(x,T)dx \\ &= (TS_L - x_L)\mathbf{U}_L + \int_{TS_L}^{TS_R} \mathbf{U}(x,T)dx + (x_R - TS_R)\mathbf{U}_R \end{aligned} \tag{8.80}$$

Comparing Eq. 8.80 with Eq. 8.79 gives

$$\int_{TS_L}^{TS_R} \mathbf{U}(x,T)dx = T\Big(S_R\mathbf{U}_R - S_L\mathbf{U}_L + \mathbf{F}_L - \mathbf{F}_R.\Big) \tag{8.81}$$

Dividing by $T(S_R - S_L)$, which is the width of the wave system of the solution of the Riemann problem between the slowest and fastest signals at time T, we have

$$\mathbf{U}^{hll} = \frac{S_R\mathbf{U}_R - S_L\mathbf{U}_L + \mathbf{F}_L - \mathbf{F}_R}{S_R - S_L}, \tag{8.82}$$

where we define the HLL state solution $\mathbf{U}^{hll}$ at T by

$$\mathbf{U}^{hll} \equiv \frac{1}{T(S_R - S_L)} \int_{TS_L}^{TS_R} \mathbf{U}(x,T)dx. \tag{8.83}$$

See Fig. 5 for the constant single vector state $\mathbf{U}^{hll}$ separated by the two fastest waves, S_L and S_R.

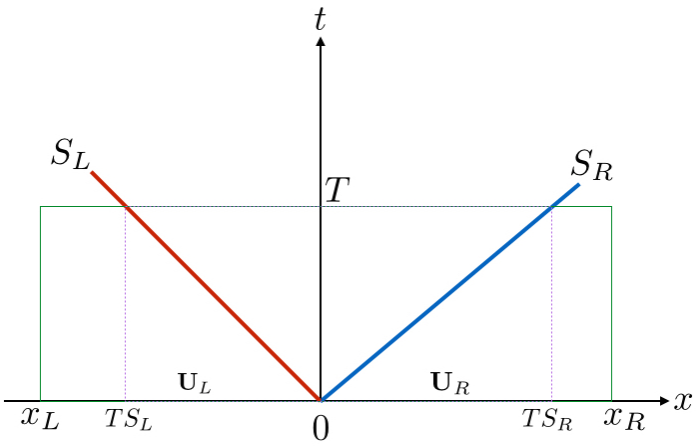


Figure 4. Control volume $[x_L, x_R] \times [0, T]$ on x-t plane. S_L and S_R are the fastest signal velocities arising from the solution of the Riemann problem.

We now apply the integral form of the conservation laws to the left portion of Fig. 4, that is, $[x_L, 0] \times [0, T]$,

$$\int_{TS_L}^{0} \mathbf{U}(x,T)dx = -TS_L\mathbf{U}_L + T(\mathbf{F}_L - \mathbf{F}_{0L}), \tag{8.84}$$

where $\mathbf{F}_{0L}$ is the flux $\mathbf{F}(\mathbf{U})$ along the t-axis. Solving for $\mathbf{F}_{0L}$ we see that

$$\mathbf{F}_{0L} = \mathbf{F}_L - S_L\mathbf{U}_L - \frac{1}{T}\int_{TS_L}^{0} \mathbf{U}(x,T)dx. \tag{8.85}$$

In the similar way, we evaluate on $[0, x_R] \times [0, T]$ and get

$$\mathbf{F}_{0R} = \mathbf{F}_R - S_R\mathbf{U}_R + \frac{1}{T}\int_{0}^{TS_R} \mathbf{U}(x,T)dx. \tag{8.86}$$

Note that if we satisfy the equality (which we should always guarantee)

$$\mathbf{F}_{0R} = \mathbf{F}_{0L} \tag{8.87}$$

then we recover the Consistency Condition Eq. 8.79.

Note: All relations derived so far are exact, as we are assuming the exact solution of the Riemann problem.

The HLL flux formulation can be put into the following approximation

$$\mathbf{U}(x,t) = \begin{cases} \mathbf{U}_L & \text{if } x/t \le S_L, \\ \mathbf{U}^{hll} & \text{if } S_L \le x/t \le S_R, \\ \mathbf{U}_R & \text{if } x/t \ge S_R, \end{cases} \tag{8.88}$$

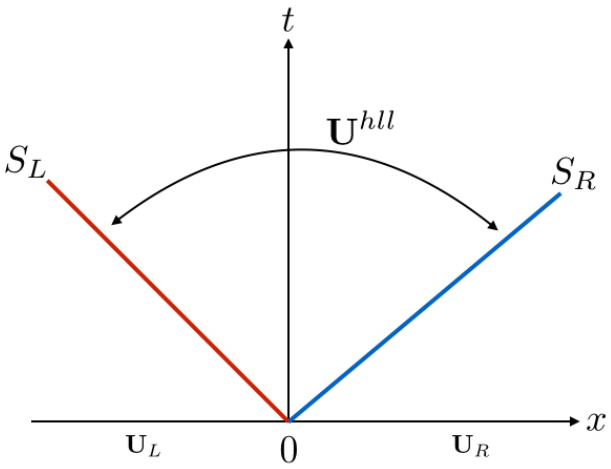


Figure 5. Approximate HLL Riemann solver. Solution in the Riemann fan region (or Star Region) consists of a *single* state $\mathbf{U}^{hll}$ separated from data states by two waves of speeds S_L and S_R.

where $\mathbf{U}^{hll}$ is the constant state vector given by Eq. 8.82, and the speeds S_L and S_R are assumed to be known. In practice, we take

$$S_L = \min\Big[(\lambda_1)_L, (\lambda_1)_R\Big], \tag{8.89}$$

$$S_R = \max\Big[(\lambda_3)_L, (\lambda_3)_R\Big], \tag{8.90}$$

where $(\lambda_k)_{L,R}$ are the 1st and 3rd eigenvalues (i.e., $k = 1$ and 3) (see Eq. 8.35) evaluated at the left and the right data states, $\mathbf{U}_{L,R}$, respectively.

Our goal is to seek for an approximated flux at each intercell boundary $x_{i+1/2}$ (or at $x = 0$ at each local frame of reference drawn as in Fig. 5) of the form

$$\mathbf{F}_{i+\frac{1}{2}} = \begin{cases} \mathbf{F}_L & \text{if } 0 \le S_L, \\ \mathbf{F}^{hll} & \text{if } S_L \le 0 \le S_R, \\ \mathbf{F}_R & \text{if } 0 \ge S_R. \end{cases} \tag{8.91}$$

The only non-trivial case of interest is therefore the subsonic case $S_L \le 0 \le S_R$. Substituting the integrand in Eq. 8.85 or Eq. 8.86 by $\mathbf{U}^{hll}$ in Eq. 8.82, we get

$$\mathbf{F}^{hll} = \mathbf{F}_L + S_L(\mathbf{U}^{hll} - \mathbf{U}_L), \tag{8.92}$$

or

$$\mathbf{F}^{hll} = \mathbf{F}_R + S_R(\mathbf{U}^{hll} - \mathbf{U}_R). \tag{8.93}$$

Note also that Eq. 8.92 and Eq. 8.93 are also derived from applying the Rankine-Hugoniot Conditions across the left and right waves respectively.

Writing out the expression of $\mathbf{U}^{hll}$ in Eq. 8.82 from Eq. 8.92 and Eq. 8.93, we finally obtain the HLL flux defined by

$$\mathbf{F}^{hll} = \frac{S_R\mathbf{F}_L - S_L\mathbf{F}_R + S_LS_R(\mathbf{U}_R - \mathbf{U}_L)}{S_R - S_L}. \tag{8.94}$$

Note: It is important to notice that we *did not take* $\mathbf{F}^{hll} = \mathbf{F}(\mathbf{U}^{hll})$, where $\mathbf{F}$ is the true flux function of the Euler equations in Eq. 8.29. This naive approach will make the resulting method unstable.

Quick summary: The HLL Riemann solver only uses two fastest characteristic waves, λ_1 and λ_3, that are genuinely nonlinear, and omits the middle wave λ_2 which is linearly degenerate. Therefore, the computed HLL Riemann solution produces more dissipative approximation in the Riemann fan region than the exact solution.

4.4.2. Roe's Approximae Riemann Solver Perhaps, the most well-known of all approximate Riemann solvers today, is the one due to Roe, which was first presented in 1981. The Roe Riemann solver is one of the most sophisticated Riemann solvers utilizing all available characteristic wave information given in the system. This is in contrast to the HLL formulation which assumes the single constant state $\mathbf{U}^{hll}$ in the Riemann fan region, thus ignoring to account for the middle wave λ_2 which corresponds to the contact discontinuity.

Roe's approach replaces the flux Jacobian matrix $\mathbf{A}(\mathbf{U})$ by a constant Jacobian matrix. The idea is the same as the one used in the linearization process of the nonlinear systems of equations, and consider the constant Roe Jacobian matrix $\bar{\mathbf{A}}$ *evaluated at a constant averaged state* $\bar{\mathbf{U}} = \mathbf{U}_{avg}$ defined as

$$\bar{\mathbf{A}} \equiv \bar{\mathbf{A}}(\mathbf{U}_{avg}) = \bar{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R) \tag{8.95}$$

and solve the approximate Riemann problem

$$\mathbf{U}_t + \bar{\mathbf{A}}\mathbf{U}_x = 0, \tag{8.96}$$

$$\mathbf{U}(x,0) = \begin{cases} \mathbf{U}_L & \text{if } x < 0, \\ \\ \mathbf{U}_R & \text{if } x > 0, \end{cases} \tag{8.97}$$

which is then solved *exactly.* (i.e., the Roe's method solves the *linearized version* Eq. 8.96 of Eq. 8.28 *exactly.*)

Let us take a moment and think about what we are doing now. The Roe's linearization process amounts to say that we replace the original nonlinear hyperbolic conservation law

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0 \tag{8.98}$$

involving with the original nonlinear (analytic) flux function $\mathbf{F}(\mathbf{U})$, with a *new modified linearized* conservation law

$$\mathbf{U}_t + \bar{\mathbf{F}}(\mathbf{U})_x = 0 \tag{8.99}$$

involving with a *new modified linearized (analytic) flux function* $\bar{\mathbf{F}}(\mathbf{U}) = \bar{\mathbf{A}}\mathbf{U}$. This new modified flux function $\bar{\mathbf{F}}(\mathbf{U})$ is presumably is easier to work with than the original flux function $\mathbf{F}(\mathbf{U})$.

Note: Notice that the property of $\bar{\mathbf{F}}(\mathbf{U})$ is now different from the original flux $\mathbf{F}(\mathbf{U})$, thus we have $\bar{\mathbf{F}}(\mathbf{U}) \neq \mathbf{F}(\mathbf{U})$ in general.

Roe suggested that the following conditions should be imposed on the Roe's linearized matrix $\bar{\mathbf{A}}$:

1. *Hyperbolicity:* $\bar{\mathbf{A}}$ has real eigenvalues that can be ordered as

$$\bar{\lambda}_1 \leq \bar{\lambda}_2 \leq \bar{\lambda}_3. \tag{8.100}$$

2. *Consistency with the exact Jacobian:*

$$\bar{\mathbf{A}}(\mathbf{U}, \mathbf{U}) = \mathbf{A}(\mathbf{U}) \tag{8.101}$$

3. *Conservation across discontinuities:*

$$\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = \bar{\mathbf{A}}(\mathbf{U}_R - \mathbf{U}_L). \tag{8.102}$$

In what follows, for the sake of simplicity, let us drop the bar ($^-$) notation in the associated eigensystem of the Roe matrix $\bar{\mathbf{A}}$ which consists of a triple $(\bar{\lambda}_k, \bar{\mathbf{r}}_k, \bar{\mathbf{l}}_k)$, and let us denote it simply by $(\lambda_k, \mathbf{r}_k, \mathbf{l}_k)$ without the bars.

Now that the modified linearized conservation system is given, we can write the initial left and right states as

$$\mathbf{U}_L = \sum_{i=1}^{3} \alpha_i \mathbf{r}_i, \tag{8.103}$$

$$\mathbf{U}_R = \sum_{i=1}^{3} \beta_i \mathbf{r}_i. \tag{8.104}$$

We have that α_k and β_k are the characteristic variables defined by the initial condition of the characteristic variable w_k given as Eq. 8.56. Using Eq. 8.25, we see that

$$\begin{aligned} \alpha_k &= \mathbf{l}_k \cdot \mathbf{U}_L \\ &= l_{k,1}\rho_L + l_{k,2}m_L + l_{k,3}E_L, \end{aligned} \tag{8.105}$$

and similarly,

$$\begin{aligned} \beta_k &= \mathbf{l}_k \cdot \mathbf{U}_R \\ &= l_{k,1}\rho_R + l_{k,2}m_R + l_{k,3}E_R. \end{aligned} \tag{8.106}$$

Here, $l_{k,i}$ denotes the i-th component of the k-th left eigenvector. For instance, from Eq. 8.37 we have

$$l_{3,1} = \frac{u^2}{2} - \frac{c_s u}{\gamma - 1}, \tag{8.107}$$

$$l_{3,2} = -u + \frac{c_s}{\gamma - 1}, \tag{8.108}$$

$$l_{3,3} = 1. \tag{8.109}$$

Substituting these expressions for α_k and β_k we can rewrite the total amount of jump $\Delta\mathbf{U}$ in terms of the left and right states,

$$\begin{aligned} \Delta\mathbf{U} &= \sum_{i=1}^{3} \Big(\beta_i - \alpha_i\Big)\mathbf{r}_i \\ &= \sum_{i=1}^{3} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\mathbf{r}_i. \end{aligned} \tag{8.110}$$

The solution of the local Riemann problems to the modified system in Eq. 8.99 at each cell interface evaluated at $x/t = 0$ is given by, from the left state,

$$\mathbf{U}_{i+\frac{1}{2}}(0) = \mathbf{U}_L + \sum_{\lambda_i \le 0} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\mathbf{r}_i, \tag{8.111}$$

which only includes the left-going k-th waves, $\lambda_k \le 0$.

Likewise, we also can get, from the right state,

$$\mathbf{U}_{i+\frac{1}{2}}(0) = \mathbf{U}_R - \sum_{\lambda_i \ge 0} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\mathbf{r}_i, \tag{8.112}$$

and we see that this only include the right-going k-th waves, $\lambda_k \ge 0$.

Our goal is now to find a *numerical flux function* associated with the modified conservation law Eq. 8.99, denoted by $\mathbf{F}^{num}_{i+1/2}$, at each cell interface $x_{i+1/2}$. With the solution of the local Riemann problem $\mathbf{U}_{i+1/2}(0)$ in Eq. 8.111 or Eq. 8.112, one obvious choice would be to define as

$$\mathbf{F}^{num}_{i+1/2} = \bar{\mathbf{A}}\mathbf{U}_{i+\frac{1}{2}}(0). \tag{8.113}$$

We however see that this choice is *not* correct if we consider a right-going supersonic flow condition, i.e., $\lambda_k > 0$ for all k. Since all the waves are biased to move from the left to the right direction, we expect the numerical flux should simply reduce to the *exact (or analytic)* flux evaluated at the left state,

$$\mathbf{F}^{num}_{i+1/2} = \mathbf{F}_L. \tag{8.114}$$

The simple relation in Eq. 8.113 fails to provide this as

$$\mathbf{F}^{num}_{i+1/2} = \bar{\mathbf{A}}\mathbf{U}_{i+\frac{1}{2}}(0) = \bar{\mathbf{A}}\mathbf{U}_L = \bar{\mathbf{F}}(\mathbf{U}_L) \neq \mathbf{F}(\mathbf{U}_L). \tag{8.115}$$

We can achieve a correct relation for $\mathbf{F}^{num}_{i+1/2}$ as follows. Since we require the solution of the local Riemann problem $\mathbf{U}_{i+\frac{1}{2}}(x,t)$ to satisfy 'the conservation law', we should expect $\mathbf{U}_{i+\frac{1}{2}}(x,t)$ to hold the conservation relation applied to $\mathbf{U}_t + \bar{\mathbf{F}}(\mathbf{U})_x = 0$ on the control volume $[TS_L, 0] \times [0, T]$, (i.e., see Eq. 8.79)

$$\int_{TS_L}^{0} \mathbf{U}_{i+\frac{1}{2}}(x,T)dx = T\Big[\bar{\mathbf{F}}(\mathbf{U}_L) - \bar{\mathbf{F}}(\mathbf{U}_{i+\frac{1}{2}}(0))\Big] - TS_L\mathbf{U}_L, \tag{8.116}$$

Likewise, on $[0, TS_R] \times [0, T]$ we get

$$\int_{0}^{TS_R} \mathbf{U}_{i+\frac{1}{2}}(x,T)dx = T\Big[\bar{\mathbf{F}}(\mathbf{U}_{i+\frac{1}{2}}(0)) - \bar{\mathbf{F}}(\mathbf{U}_R)\Big] + TS_R\mathbf{U}_R, \tag{8.117}$$

At the same time, from the relation in Eq. 8.85 or Eq. 8.86, we have

$$\mathbf{F}_{0L} = \mathbf{F}_L - S_L\mathbf{U}_L - \frac{1}{T}\int_{TS_L}^{0} \mathbf{U}_{i+\frac{1}{2}}(x,T)dx, \tag{8.118}$$

or

$$\mathbf{F}_{0R} = \mathbf{F}_R - S_R\mathbf{U}_R + \frac{1}{T}\int_{0}^{TS_R} \mathbf{U}_{i+\frac{1}{2}}(x,T)dx. \tag{8.119}$$

Finally, if we combine Eq. 8.116 and Eq. 8.118, we obtain

$$\mathbf{F}_{0L} = \mathbf{F}_L - \bar{\mathbf{F}}(\mathbf{U}_L) + \bar{\mathbf{F}}(\mathbf{U}_{i+\frac{1}{2}}(0)). \tag{8.120}$$

Similarly, combining Eq. 8.117 and Eq. 8.119, we get

$$\mathbf{F}_{0R} = -\bar{\mathbf{F}}(\mathbf{U}_R) + \bar{\mathbf{F}}(\mathbf{U}_{i+\frac{1}{2}}(0)) + \mathbf{F}_R. \tag{8.121}$$

Now if we use the definition of the flux $\bar{\mathbf{F}} = \bar{\mathbf{A}}\mathbf{U}$ applied to Eq. 8.120 and Eq. 8.120 respectively, and use the fact

$$\bar{\mathbf{A}}\mathbf{r}_k = \lambda_k\mathbf{r}_k, \tag{8.122}$$

we get

$$\begin{aligned}
\mathbf{F}_{0L} &= \mathbf{F}_L - \bar{\mathbf{F}}(\mathbf{U}_L) + \bar{\mathbf{F}}(\mathbf{U}_{i+\frac{1}{2}}(0)) && (8.123)\\
&= \mathbf{F}_L - \bar{\mathbf{A}}\mathbf{U}_L + \bar{\mathbf{A}}\Big[\mathbf{U}_L + \sum_{\lambda_i \le 0} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\mathbf{r}_i\Big] && (8.124)\\
&= \mathbf{F}_L + \sum_{\lambda_i \le 0} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\lambda_i\mathbf{r}_i && (8.125)
\end{aligned}$$

In the same manner, we obtain

$$\mathbf{F}_{0R} = \mathbf{F}_R - \sum_{\lambda_i \ge 0} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)\lambda_i\mathbf{r}_i. \tag{8.126}$$

For consistency, we require

$$\mathbf{F}_{0L} = \mathbf{F}_{0R} \tag{8.127}$$

and set to be

$$\mathbf{F}^{num}_{i+\frac{1}{2}} \equiv \mathbf{F}_{0L} = \mathbf{F}_{0R} \tag{8.128}$$

Alternatively, we can sum Eq. 8.123 and Eq. 8.126 and get an averaged expression

$$\mathbf{F}^{num}_{i+\frac{1}{2}} = \frac{1}{2}\Big(\mathbf{F}_L + \mathbf{F}_R\Big) - \frac{1}{2}\sum_{i=1}^{3} \mathbf{l}_i \cdot \Big(\mathbf{U}_R - \mathbf{U}_L\Big)|\lambda_i|\mathbf{r}_i. \tag{8.129}$$

Note: We note here that $\mathbf{F}_{L,R}$ are the fluxes evaluated exactly using the analytic Euler flux formulation in Eq. 8.29 at $\mathbf{U}_{L,R}$ respectively.

Note: The eigensystem $(\lambda_k, \mathbf{r}_k, \mathbf{l}_k)$ is to be evaluated analytically using the formulas for the *conservative ones* in Eq. 8.35, Eq. 8.36, and Eq. 8.37, evaluated at an averaged state $\mathbf{U}_{avg}$.

Note: We can in practice take an arithmetic average to obtain $\mathbf{U}_{avg}$,

$$\mathbf{U}_{avg} = \frac{1}{2}\Big(\mathbf{U}_L + \mathbf{U}_R\Big). \tag{8.130}$$

However, this simple averaging in general does not guarantee to satisfy the conservation property in Eq. 8.102 of the Roe matrix. Although in practice, it turns out that the simple arithmetic works pretty well. One can improve this situation by considering so-called the *Roe averages*.

Quick summary: It should be noted that the Roe solver include all available wave characteristics in its formulation. Hence it resolves better Riemann solutions at the Riemann fan and produces sharper resolutions compared to the HLL solutions which lacks the contact discontinuity solution.

Quick summary: Remind that we have used the following different entities:

- $\mathbf{F}$ is the exact (or analytic) flux function for the original nonlinear conservation laws,
- $\bar{\mathbf{F}}(\mathbf{U}) = \bar{\mathbf{A}}\mathbf{U}$ is the exact (or analytic) flux function for the modified linearized conservation laws,
- $\mathbf{F}^{num}_{i+1/2}$ is the numerical flux function of the modified linearized conservation laws, and
- $\mathbf{U}_{i+1/2}(x,t)$ is the solution of the local Riemann problem of the modified linearized conservation laws.

Chapter 9

Finite Volume Reconstruction Schemes for FOG, PLM, PPM and WENO

Important Remark on Notational Changes:

In what follows we are making a few changes in writing the k-th wave family so that the new notations can bear a clearer description of the local cell index i.

We write the wave family "k" using a superscript with parentheses and the cell index i as a subscript. For instance, we replace the k-th eigensystem triple of the eigenvalues, the left and the right eigenvectors

$$(\lambda_k, \mathbf{r}_k, \mathbf{l}_k) \tag{9.1}$$

with

$$(\lambda_i^{(k)}, \mathbf{r}_i^{(k)}, \mathbf{l}_i^{(k)}) \tag{9.2}$$

which are all evaluated at each point x_i in a local cell $I_i = [x_{i-1/2}, x_{i+1/2}]$. When needed, we also differentiate the conservative and primitive eigenvectors by denoting them, for instance, as

$$\mathbf{l}^{(k,c)}, \mathbf{l}^{(k,p)}, \tag{9.3}$$

respectively. We may drop the superscript by conveniently assuming a correct choice of eigenvectors applied to a given variable under consideration, i.e., conservative or primitive.

We also utilize general component forms at each cell I_i of the discrete numerical data of the primitive ($\mathbf{V}$), conservative ($\mathbf{U}$) and characteristic ($\mathbf{W}$) variables denoted as

$$\mathbf{V}_i = \begin{pmatrix} v_{i:1} \\ v_{i:2} \\ v_{i:3} \end{pmatrix} = \begin{pmatrix} v_{i:\rho} \\ v_{i:u} \\ v_{i:p} \end{pmatrix} = \begin{pmatrix} \rho_i \\ u_i \\ p_i \end{pmatrix}, \tag{9.4}$$

$$\mathbf{U}_i = \begin{pmatrix} u_{i:1} \\ u_{i:2} \\ u_{i:3} \end{pmatrix} = \begin{pmatrix} u_{i:\rho} \\ u_{i:m} \\ u_{i:E} \end{pmatrix} = \begin{pmatrix} \rho_i \\ m_i \\ (\rho E)_i \end{pmatrix}, \tag{9.5}$$

$$\mathbf{W}_i = \begin{pmatrix} w_i^{(1)} \\ w_i^{(2)} \\ w_i^{(3)} \end{pmatrix} = \begin{pmatrix} \mathbf{l}_i^{(1,c)} \cdot \mathbf{U}_i \\ \mathbf{l}_i^{(2,c)} \cdot \mathbf{U}_i \\ \mathbf{l}_i^{(3,c)} \cdot \mathbf{U}_i \end{pmatrix} = \begin{pmatrix} \mathbf{l}_i^{(1,p)} \cdot \mathbf{V}_i \\ \mathbf{l}_i^{(2,p)} \cdot \mathbf{V}_i \\ \mathbf{l}_i^{(3,p)} \cdot \mathbf{V}_i \end{pmatrix}. \tag{9.6}$$

Often we would drop the index notation for components and simply represent *each* component by u_i, v_i and w_i.

1. Reconstruction Schemes

High-order reconstruction schemes for 1D problems are formulated consisting of high-order state predictions in normal direction accommodating solution accuracy (*viz.* normal prediction step). In this section, we provide an overall numerical description on how to calculate the normal prediction in five different types of reconstruction schemes for the Euler equations:

- the first-order Godunov method (Godunov, 1961),
- the second-order piecewise linear method (PLM) (Colella, 1985; LeVeque; Toro),
- the third-order piecewise parabolic (PPM) method by Colella and Woodward (Colella & Woodward, 1984),
- the fifth-order weighted essentially non-oscillatory (WENO-5) by Jiang and Shu, 1996, and
- the fifth-order WENO-Z method by Borges et al. (2008) as a variant of WENO-5.

In the following, we provide concrete REA formulations of the five different normal prediction algorithms that provide differing orders of *spatial* accuracy, while keeping the *temporal* accuracy. Overall, this approach is second-order accurate in space and time which belongs to a single-step explicit predictor-corrector formalism.

Before discussing mathematical algorithms for reconstruction, we first note that there are two fundamental principles in formulating such a class of reconstruction algorithms in FVM.

1.1. Two Basic Principles

Two very important basic principles that a class of reconstruction algorithms require to satisfy in the normal prediction step include:
(i) monotonic high-order spatial reconstruction – predictor, and
(ii) half-time step temporal evolution – corrector.

For exposition purposes together with the fact that we are most interested in solving 1D systems, let us consider the above two properties only in x-direction (which is the only normal direction in 1D).

1.1.1. First Principle: Monotonic Reconstruction The first principle is to design a monotone-preserving reconstruction by choosing an approximating polynomial $p_i = p_i(x)$ of degree n on each i-th cell $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, whose shape of reconstruction on I_i preserves the cell volume-averaged quantity of each variable 'v' (preferably primitive variable in practice) under consideration. That is, p_i is chosen such that it satisfies

$$\bar{p}_i^n = \bar{v}_i^n, \tag{9.7}$$

where

$$\bar{p}_i^n = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(x, t^n)\, dx, \tag{9.8}$$

and

$$\bar{v}_i^n = \frac{1}{\Delta x_i} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} v(x, t^n)\, dx\,. \tag{9.9}$$

This constraint is very important in a sense that the approximating polynomial does not lose one of the key properties in FVM in as much as FVM always evolves the integral quantities, i.e., volume-averaged quantities over each cell.

The choice of the polynomial

$$p_i(x) = \sum_{k=0}^{n} c_k (x - x_i)^k \tag{9.10}$$

with $n \geq 0$ naturally determines the 1D reconstruction scheme in normal direction to be an $(n+1)$-th order accurate formulation in space.

Once $p_i(x)$ is chosen, the next step is to obtain cell edge nodal values $v_{L,i}^n$ and $v_{R,i}^n$. These nodal values are easily available by directly calculating

$$v_{L,i}^n \equiv p_i(x_{i-\frac{1}{2}}), \text{ and } v_{R,i}^n \equiv p_i(x_{i+\frac{1}{2}}). \tag{9.11}$$

Although the way we obtain Eq. 9.11 is straightforward once $p_i(x)$ is known, there is one remaining set of constraints that needs to be carried out on $v_{L,i}^n$ and $v_{R,i}^n$, in order to preserve non-oscillatory, monotonic states of the two reconstructed states at the cell edges. This procedure is often called a 'limiting' step which checks the following two conditions:

- *Condition 1:* The profile of $p_i(x)$ must be monotonic over I_i. See the left panel in Fig. 1.
- *Condition 2:* $v_{L,i}^n$ and $v_{R,i}^n$ should lie between the neighboring volume-averaged quantities, $\bar{v}_{i\pm1}^n$. In other words, one should guarantee $v_{L,i}^n \in [\min(\bar{v}_{i-1}^n, \bar{v}_i^n), \max(\bar{v}_{i-1}^n, \bar{v}_i^n)]$ and $v_{R,i}^n \in [\min(\bar{v}_i^n, \bar{v}_{i+1}^n), \max(\bar{v}_i^n, \bar{v}_{i+1}^n)]$ See the right panel in Fig. 1.

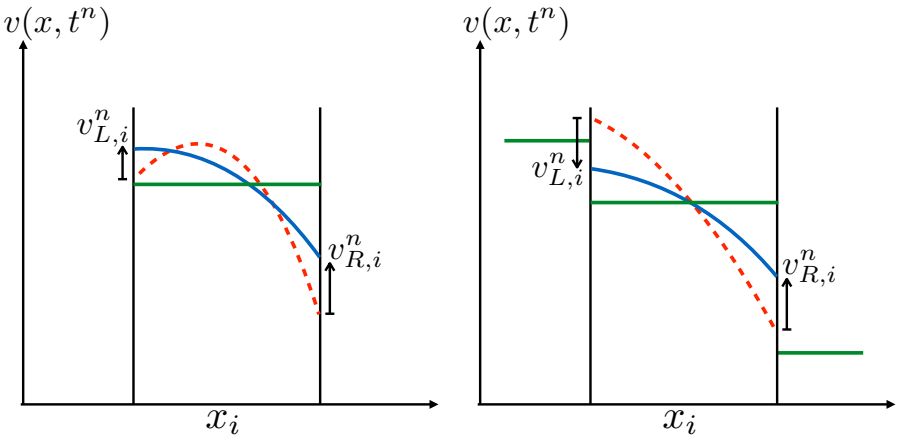


Figure 1. The correction procedure for monotonic reconstruction in the case of PPM. In both figures, the red dotted line represents the uncorrected profile of $p_i(x)$; the solid blue line indicates the corrected profile; the direction of the arrows denotes newly corrected edge values; and the green solid line illustrates the cell-averaged quantities $\bar{v}^n_k$ on I_k. Left: *Condition 1* checks if the profile of the reconstructed polynomial function $p_i(x)$ on I_i is monotonic. In this case, the original profile (red) of $p_i(x)$ has produced a new local maximum, for which a new monotone-preserving profile (blue) needs to be sought. The new corrected edge values are then recomputed from the new profile. Right: *Condition 2* ensures that the newly calculated nodal values $v^n_{L;R,i}$ at cell edges $x_{i\pm1/2}$ are bounded by the cell-averaged values $\bar{v}^n_{i;i\pm1}$ in the neighboring cells. The original profile (red) has generated $v^n_{L,i}$ which is larger than $\bar{v}^n_{i-1}$ (green). A correction to get a new profile $p_i(x)$ (blue) is needed in order to lower $v^n_{L,i}$ that is bounded by $\bar{v}^n_{i-1}$ and $\bar{v}^n_i$. And at the same time, the new profile must bound $v^n_{R,i}$ between $\bar{v}^n_i$ and $\bar{v}^n_{i+1}$ as well.

If any one of the above conditions is not satisfied, $v^n_{L,i}$ and $v^n_{R,i}$ need to be corrected by adjusting a new profile $p_i(x)$ to meet the two conditions; otherwise numerical solutions using the reconstructed states can lead to erroneous oscillations especially at discontinuities. It is important that any newly corrected $p_i(x)$ must satisfy the relation in Eq. (9.7) throughout the correction procedure.

1.1.2. Second Principle: Half-Time Step Evolution The second principle is to perform so-called a 'half-time step' evolution that advances the reconstructed edge states $v^n_{L;R,i}$ by $\Delta t/2$ to achieve $v^{n+\frac{1}{2}}_{L;R,i}$. The main idea, which was put forward by van Leer and Hancock, and studied in the algorithm, named as MUSCL-Hancock (MH), is to enhance temporal accuracy to second-order by means of computing Godunov fluxes at half time step, following the mid-point method in ODEs.

Note that at each cell x_i, we have a 1D characteristic equation in each normal direction (x-direction in our case) for each k-th wave,

$$\frac{\partial w_i^{(k)}}{\partial t} + \lambda_i^{(k)} \frac{\partial w_i^{(k)}}{\partial x} = 0, \tag{9.12}$$

where $w_i^{(k)}$ is a k-th component of $\mathbf{W}_i$. The relationship between the primitive variables $\mathbf{V}_i$ and $\mathbf{W}_i$ is given by projecting one from the other via the eigenvectors,

$$\mathbf{V}_i = \sum_k \mathbf{r}_i^{(k)} w_i^{(k)}, \text{ where } w_i^{(k)} = \mathbf{l}_i^{(k)} \cdot \mathbf{V}. \tag{9.13}$$

Since the linear combination in Eq. (9.13) also holds for $p_i(x)$ on I_i, we can compute a half-time step advancement of an m-th reconstructed component $v_{i:m}$ of $\mathbf{V}_i$ on I_i (i.e., $v_{i:m} = \mathbf{V}_i \cdot \mathbf{e}_m$) at the right edge $x = x_{i+1/2}$ as

$$v_{R,i:m}^{n+\frac{1}{2}} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} p_i(x_{i+\frac{1}{2}}, t)\, dt = \sum_k \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} r_{i:m}^{(k)} w_i^{(k)}(x_{i+\frac{1}{2}}, t)\, dt, \tag{9.14}$$

where $r_{i:m}^{(k)} = \mathbf{r}_i^{(k)} \cdot \mathbf{e}_m$ is an m-th projection evaluated on I_i with the unit vector $\mathbf{e}_m$.

Now let $\mathbf{P}_i$ be a vector consisting of the individual reconstructed profiles $p_{i:m}(x)$ on I_i for each m-th primitive variable $v_{i:m}$ of $\mathbf{V}_i$ which is of length 3 in our hydrodynamics case, that is,

$$\mathbf{P}_i = (p_{i:1}, p_{i:2}, p_{i:3})^T = (p_{i:\rho}, p_{i:u}, p_{i:p})^T. \tag{9.15}$$

The exact solution of $w_i^{(k)}(x_{i+\frac{1}{2}}, t)$ on I_i at $t = t^n + \Delta t$ (i.e., the farthest thinner blue curve in Fig. 2) is found by tracing back in time to t^n along its characteristic line $\frac{dx}{dt} = \lambda_i^{(k)}$, where the spatial reconstruction of $w_i^{(k)}(x_{i+\frac{1}{2}}, t^n)$ (i.e., the near thicker blue curve in Fig. 2) is readily available via the reconstructed profiles $\mathbf{P}_i(x)$. Thus for each right-going wave with a positive characteristic velocity $\lambda_i^{(k)} > 0$ that reaches to the right edge at $x_{i+\frac{1}{2}}$, we get

$$\begin{aligned} w_i^{(k)}(x_{i+\frac{1}{2}}, t^n + \Delta t) &= w_i^{(k)}(x_{i+\frac{1}{2}} - \lambda_i^{(k)} \Delta t) \\ &= \mathbf{l}_i^{(k)} \cdot \mathbf{V}_i(x_{i+\frac{1}{2}} - \lambda_i^{(k)} \Delta t) \\ &= \mathbf{l}_i^{(k)} \cdot \mathbf{P}_i(x_{i+\frac{1}{2}} - \lambda_i^{(k)} \Delta t). \end{aligned} \tag{9.16}$$

The value $x_{i+\frac{1}{2}} - \lambda_i^{(k)} \Delta t$ is the 'foot' of the characteristic line $\frac{dx}{dt} = \lambda_i^{(k)}$, where the line intersects with the x-axis at $t = t^n$ in the x-t phase space. The characteristic variable $w_i^{(k)}$ is invariant along the characteristic line in the phase space, which we used in the first equality in Eq. (9.16). See Fig. 2. Here $\lambda_i^{(k)}$ and $\mathbf{l}^{(k)}$ denote an eigenvalue and left eigenvector respectively, both corresponding to the k-th wave and evaluated on I_i.

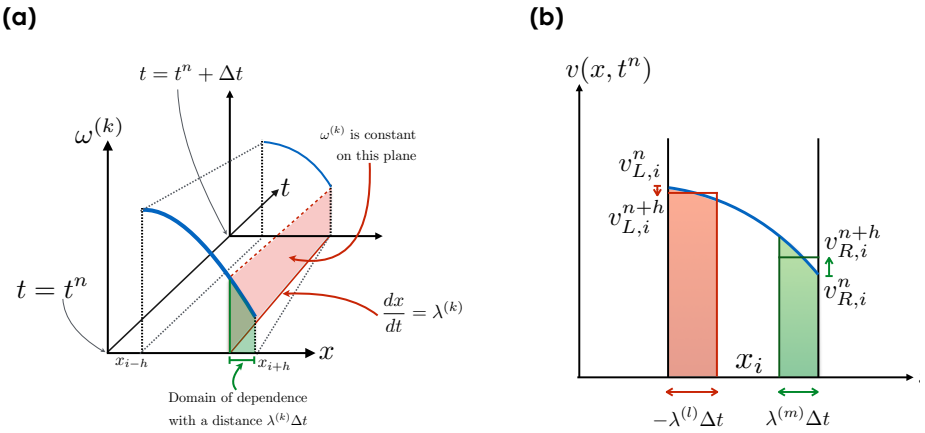


Figure 2. The shorthand notation 'h' is used to represent the half-index $\frac{1}{2}$ in the above illustrations. (a) The characteristic tracing procedure from $t^n + \Delta t$ to t^n. The reconstructed profiles of the k-th characteristic variable $\omega^{(k)}$ at the two different time steps are illustrated in blue thick curves. The farthest thinner blue curve represents a not-yet reconstructed profile of $\omega^{(k)}(x, t = t^n + \Delta t)$, whereas the near thicker blue curve shows an already reconstructed profile of $\omega^{(k)}(x, t = t^n)$, both on the cell I_i. The characteristic line with a positive characteristic velocity $\lambda^{(k)} > 0$ is drawn in solid red line, showing a backtracing in time to t^n. The invariance property of $\omega^{(k)}$ is preserved on the red-shaded plane, of which upper boundary in red-dotted line denotes its constant value over time. The green-shaded area represents the contribution of the k-th wave to $v^{n+\frac{1}{2}}_{R,i}$, which is an integrated average quantity of $\omega^{(k)}(x, t = t^n)$ over a domain of dependence $[x_{i+\frac{1}{2}} - \lambda^{(k)}\Delta t, x_{i+\frac{1}{2}}]$ at $t = t^n$ for the k-th characteristic. (b) Half-time step advancements of the spatially reconstructed left and right states $v^n_{L;R,i}$ to $v^{n+\frac{1}{2}}_{L;R,i}$ on cell I_i by tracing relavant characteristics. For the left state, the tracing only involves with any left-going l-th characteristics (i.e., $\lambda^{(l)} < 0$ in this case). The area of the red-shaded region represents an averaged quantity over the resultant domain of dependence from the n-th characteristic. This area amounts how much contributions could be carried out to change $v^n_{L,i}$ over $\Delta t/2$ by the n-th characteristic. The area of the rectangle in red line is equal to that of the red-shaded region, showing the new evolved state $v^{n+\frac{1}{2}}_{L,i}$. The similar is true for the right state with the right-going m-th characteristic as illustrated.

Then the integration in Eq. (9.14) becomes,

$$\begin{aligned} v_{R,i:m}^{n+\frac{1}{2}} &= \sum_k \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \mathbf{P}_i(x_{i+\frac{1}{2}} - \lambda_i^{(k)}(t-t^n))\, dt, \\ &= \sum_{k;\lambda_i^{(k)}>0} \frac{1}{\lambda_i^{(k)}\Delta t} \int_{x_{i+\frac{1}{2}}-\lambda_i^{(k)}\Delta t}^{x_{i+\frac{1}{2}}} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \mathbf{P}_i(x)\, dx\,. \end{aligned} \tag{9.17}$$

We see that a new temporally-evolved, spatially-averaged right state value $v_{R,i:m}^{n+\frac{1}{2}}$ on I_i is successfully achieved by tracing back each k-th characteristic to the old state $t = t^n$, taking integrations to obtain averaged reconstructed profiles over each domain of dependence, and taking a sum of them over all characteristics that reach to the cell interface at $x_{i+\frac{1}{2}}$. See also Fig. 2.

Similarly, we get for the left state,

$$\begin{aligned} v_{L,i:m}^{n+\frac{1}{2}} &= \sum_k \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \mathbf{P}_i(x_{i-\frac{1}{2}} - \lambda_i^{(k)}(t-t^n))\, dt, \\ &= \sum_{k;\lambda_i^{(k)}<0} \frac{1}{\lambda_i^{(k)}\Delta t} \int_{x_{i-\frac{1}{2}}-\lambda_i^{(k)}\Delta t}^{x_{i-\frac{1}{2}}} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \mathbf{P}_i(x)\, dx\,. \end{aligned} \tag{9.18}$$

Quick summary: So far, we have discussed the two basic ingredients in building high-order reconstructions. The first principle is to formulate a spatially monotonic reconstruction profile $p_i(x)$ of order k on each cell I_i. The second principle is to use the characteristic tracing in order to advance the two left and right reconstructed state values computed from $p_i(x)$ at $t = t^n$ by half a time $t = t^{n+\frac{1}{2}}$. This completes our normal prediction step in each normal direction. The pair $(v_L, v_R) = (v_{R,i}^{n+\frac{1}{2}}, v_{L,i+1}^{n+\frac{1}{2}})$ of these normal predicted values at each interface $x_{i+\frac{1}{2}}$ comprises of the initial left and right states for the local Riemann problems. Upon solving the local Riemann problems, we compute the upwind Godunov fluxes, and use them to evolve conservative variables $\mathbf{U}^n$ to $\mathbf{U}^{n+1}$.

1.2. First-order Godunov's Method

The first successful numerical method for nonlinear conservative hyperbolic systems became available by Godunov in 1959. This scheme is an extension of its predecessor for solving scalar conservation laws by choosing upwind directions, often called the CIR method (see Eq. 6.34), developed by Courant, Isaacson and Rees (hence the name).

The key idea of the first-order Godunov's scheme (FOG for short) is to seek for the solution of the Riemann problem, either by an exact or an approximate solver, using a constant (or flat) reconstruction. In other words, the choice of the polynomial in Eq. (9.10) is simply given as, on I_i,

$$p_i(x) = c_0. \tag{9.19}$$

Obviously, the flat reconstruction profile satisfies all of the monotonicity conditions in the first principle. Also, the second principle of the half-time advancement step doesn't change anything but yields

$$v_{L;R,i}^{n+\frac{1}{2}} = v_{L;R,i}^{n}. \tag{9.20}$$

1.3. Second-order Piecewise Linear Method

The key ingredient in the second-order approach is to use a linear reconstruction profile

$$p_i(x) = c_0 + c_1(x - x_i), \tag{9.21}$$

which can overcome the downside of FOG - practically unsuitable for any real application problems by being too diffusive.

1.3.1. Step 1: Linear Profile Applying the relationship in Eq. (9.7) to the linear piecewise polynomial $p_i(x)$ on I_i, we obtain

$$c_0 = \bar{v}_i^n \text{ and } c_1 = \frac{\Delta v_i^n}{\Delta x}, \tag{9.22}$$

where Δv_i^n is a properly chosen slope in order for the resulting $p_i(x)$ to satisfy both *Condition 1* and *Condition 2* of the first principle. The quantity Δv_i^n is going to be determined shortly. The two states at the left and right interfaces $x_{i\pm 1/2}$ in Eq. (9.11) are then given by

$$v_{L,i}^n = \bar{v}_i^n - \frac{\Delta v_i^n}{2}, \text{ and } v_{R,i}^n = \bar{v}_i^n + \frac{\Delta v_i^n}{2}. \tag{9.23}$$

1.3.2. Step 2: Characteristic Tracing We continue to proceed the characteristic tracing in Eqs. (9.17) and (9.18). Considering the integrand of Eq. (9.17) for the right state, we get

$$\begin{aligned}
&r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot (p_{i:1}, p_{i:2}, p_{i:3})^T \\
&= r_{i:m}^{(k)} \sum_{s=1}^{3} l_{i:s}^{(k)} \left(\bar{v}_{i:s}^n + (x - x_i) \frac{\Delta v_{i:s}^n}{\Delta x} \right) \\
&= r_{i:m}^{(k)} \sum_{s=1}^{3} l_{i:s}^{(k)} \bar{v}_{i:s}^n + \frac{(x - x_i)}{\Delta x} r_{i:m}^{(k)} \sum_{s=1}^{3} l_{i:s}^{(k)} \Delta v_{i:s}^n,
\end{aligned} \tag{9.24}$$

where $r_{i:m}^{(k)} = \mathbf{r}_i^{(k)} \cdot \mathbf{e}_m$ and $l_{i:m}^{(k)} = \mathbf{l}_i^{(k)} \cdot \mathbf{e}_m$ are the m-th projections of the right and left eigenvectors evaluated on I_i with the unit vector $\mathbf{e}_m$.

Since the first term is constant, its integration in Eq. (9.17) is just

$$\sum_{k;\lambda_i^{(k)}>0} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \bar{\mathbf{V}}_i^n. \tag{9.25}$$

Although this constant term only includes the positive waves that propagates towards the right interface, in the case of using the HLL solver, we also include

a correction for waves which travel away from the interface with negative wave speeds. This is to replace Eq.(9.25) with

$$\sum_{k=1}^{3} r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \bar{\mathbf{V}}_i^n, \tag{9.26}$$

in order to keep the minimum accuracy in the cell volume averages. This correction procedure is not needed when the Roe solver is used for simulations. By this way, in particular, we retain the full first-order accuracy provided by the constant cell-averaged state $\bar{\mathbf{V}}_i^n$ when $\Delta v_{i:h}^n = 0$, and the PLM scheme reduces to FOG.

The integration in Eq. (9.17) of the second term in Eq. (9.24) becomes

$$\frac{1}{2} \sum_{k;\lambda_i^{(k)}>0} \Big(1 - \frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big) r_{i:m}^{(k)} \Delta w_i^{(k)} \tag{9.27}$$

where

$$\Delta w_i^{(k)} = \sum_{s=1}^{3} l_{i:s}^{(k)} \Delta v_{i:s}^n = \mathbf{l}^{(k)} \cdot \Delta \mathbf{V}_i^n. \tag{9.28}$$

Note here that $\Delta w_i^{(k)}$ is a jump across the k-th characteristic, and is often called as *k-th characteristic slope limiting*, generally combined with a monotone-preserving TVD slope limiter,

$$\Delta w_i^{(k)} = \text{TVD\_limiter}\Big[\mathbf{l}_i^{(k)} \cdot (\mathbf{V}_{i+1}^n - \mathbf{V}_i^n), \mathbf{l}_i^{(k)} \cdot (\mathbf{V}_i^n - \mathbf{V}_{i-1}^n)\Big]. \tag{9.29}$$

The choice of TVD_limiter can be one of the limiters of minmod, van Leer's, and MC in Eq. (7.13). The two conditions in the first principle are also guaranteed to hold using these limiters.

Another variant in calculating slopes is feasible, which is to first apply TVD slope limiters to the primitive variables and then to take projections of them with left eigenvectors. This is referred to as *primitive limiting* which takes of a form,

$$\Delta w_i^{(k)} = \mathbf{l}_i^{(k)} \cdot \text{TVD\_limiter}\Big[\mathbf{V}_{i+1}^n - \mathbf{V}_i^n, \mathbf{V}_i^n - \mathbf{V}_{i-1}^n\Big]. \tag{9.30}$$

When comparing the two limiting approaches, the characteristic limiting is a preferred way to better maintain monotonicity. This is simply because it is the characteristic variable that the invariant property holds along each characteristic line, during the characteristic tracing procedure.

Putting all things together, we achieve the half-time evolved normal predicted states for the second-order piecewise linear method in vector expression,

$$\mathbf{V}_{R,i}^{n+\frac{1}{2}} = \bar{\mathbf{V}}_i^n + \frac{1}{2} \sum_{k;\lambda_i^{(k)}>0} \Big(1 - \frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big) \mathbf{r}_i^{(k)} \Delta w_i^{(k)}, \tag{9.31}$$

and

$$\mathbf{V}_{L,i}^{n+\frac{1}{2}} = \bar{\mathbf{V}}_i^n + \frac{1}{2}\sum_{k;\lambda_i^{(k)}<0}\Big(-1-\frac{\lambda_i^{(k)}\Delta t}{\Delta x}\Big)\mathbf{r}_i^{(k)}\Delta w_i^{(k)}. \tag{9.32}$$

1.4. Third-order Piecewise Parabolic Method

The extension to a third-order method based on a polynomial

$$p_i(x) = c_0 + c_1(x-x_i) + c_2(x-x_i)^2 \tag{9.33}$$

is now described. This PPM method is proposed by Colella and Woodward (1984), which has been by far, one of the most popular reconstruction schemes over three decades. Unlike the second-order PLM scheme, where there is no distinction between the cell-centered nodal point v_i^n and the cell volume-averaged quantity $\bar{v}_i^n$, the PPM method clearly distinguishes them and use them differently. The term 'reconstruction', in this sense, refers to a procedure to build *nodal* quantities $v_{i\pm1/2}^n$ (in particular left and right states at interfaces) from the given *cell-averaged volume* quantities $\bar{v}_i^n$.

1.4.1. Step 1: Parabolic Profile We begin to specify three different conditions in order to determine three unknowns c_k, $k = 0, 1, 2$. The first condition is obvious - Eq. (9.7), from which we obtain

$$\bar{v}_i^n = c_0 + \frac{c_2}{12}\Delta x^2. \tag{9.34}$$

The other two conditions are readily available *if* we further assume we know how to compute the two nodal values $v_{L;R,i}$. With the help of these two values, we get

$$v_{L;R,i}^n = p_i(x_{i\pm1/2}) = c_0 \pm \frac{c_1}{2}\Delta x + \frac{c_2}{4}\Delta x^2. \tag{9.35}$$

Determining the unknowns, we achieve

$$c_0 = \bar{v}_i^n - \frac{c_2}{12}\Delta x^2, \tag{9.36}$$

$$c_1 = \frac{1}{\Delta x}(v_{R,i}^n - v_{L,i}^n), \tag{9.37}$$

$$c_2 = \frac{6}{\Delta x^2}\Big(\frac{v_{R,i}^n + v_{L,i}^n}{2} - \bar{v}_i^n\Big). \tag{9.38}$$

Therefore, the piecewise parabolic polynomial $p_i(x)$ for PPM is ready to be determined completely if we find $v_{L;R,i}^n$, for which we proceed to use separate third-degree polynomials $\phi_\pm(x)$,

$$\phi_\pm(x) = \sum_{k=0}^{3} a_k^\pm (x - x_{i\pm1/2})^k. \tag{9.39}$$

In order to determine this third-degree polynomial, we carry out considering $\phi_\pm(x)$ over four different adjacent cells, centering $v_{L;R,i}^n$ in a symmetric fashion.

In this way, we consider $\phi_-(x)$ over $I_{i-2}, I_{i-1}, I_i, I_{i+1}$ for $v^n_{L,i}$, whereas $\phi_+(x)$ over $I_{i-1}, I_i, I_{i+1}, I_{i+2}$ for $v^n_{R,i}$, on each of which, $\phi_\pm(x)$ satisfies

$$\frac{1}{\Delta x}\int_{I_k}\phi_-(x)\,dx = \bar{v}^n_k, \text{ for } i-2 \le k \le i-1, \tag{9.40}$$

and

$$\frac{1}{\Delta x}\int_{I_k}\phi_+(x)\,dx = \bar{v}^n_k, \text{ for } i-1 \le k \le i+2. \tag{9.41}$$

After a bit of algebra, we can obtain the coefficients $a^\pm_k$, with $s=1$ for a^+_k, while $s=0$ for a^-_k,

$$a^\pm_0 = \frac{1}{12}\Big(-\bar{v}^n_{i-2+s} + 7\bar{v}^n_{i-1+s} + 7\bar{v}^n_{i+s} - \bar{v}^n_{i+1+s}\Big), \tag{9.42}$$

$$a^\pm_1 = \frac{1}{12\Delta x}\Big(\bar{v}^n_{i-2+s} - 15\bar{v}^n_{i-1+s} + 15\bar{v}^n_{i+s} - \bar{v}^n_{i+1+s}\Big), \tag{9.43}$$

$$a^\pm_2 = \frac{1}{4\Delta x^2}\Big(\bar{v}^n_{i-2+s} - \bar{v}^n_{i-1+s} - \bar{v}^n_{i+s} + \bar{v}^n_{i+1+s}\Big), \tag{9.44}$$

$$a^\pm_3 = \frac{1}{6\Delta x^3}\Big(-\bar{v}^n_{i-2+s} + 3\bar{v}^n_{i-1+s} - 3\bar{v}^n_{i+s} + \bar{v}^n_{i+1+s}\Big). \tag{9.45}$$

The two nodal values $v^n_{L;R,i}$ now easily follow via the reconstruction polynomials,

$$v^n_{L,i} = \phi_-(x_{i-\frac{1}{2}}) = a^-_0, \text{ and } v^n_{R,i} = \phi_+(x_{i+\frac{1}{2}}) = a^+_0. \tag{9.46}$$

Compared to the case of PLM in Eq. (9.23), in which a pair of the left and right states at $(x_{i+\frac{1}{2}}, t^n)$ doesn't need to be continuous (i.e., $v^n_{R,i} \neq v^n_{L,i+1}$), the PPM states given by Eq. (9.46) are continuous at the interface, $v^n_{R,i} = v^n_{L,i+1}$, because $\phi_\pm(x)$ are continuous over the four adjacent cells.

It may sound weird to have such a pair of continuous left and right states at every interface, especially when the pair is to be used for the Riemann problem. Any continuous pair of Riemann states will simply not produce any flux that amounts to flow across the interface. However, this continuity at $t = t^n$ are no longer to be true when the next step of characteristic tracing evolves the n-states to $n+\frac{1}{2}$, which results in $v^{n+\frac{1}{2}}_{R,i} \neq v^{n+\frac{1}{2}}_{L,i+1}$.

The expression of $a^\pm_0$ in Eq. (9.42) can be put into another form using slope limiters, which helps to keep monotone profiles better in reconstruction,

$$a^\pm_0 = \frac{1}{2}\Big(\bar{v}^n_{i-1+s} + \bar{v}^n_{i+s}\Big) - \frac{1}{6}\Big(\Delta\bar{v}^n_{i+s} - \Delta\bar{v}^n_{i-1+s}\Big), \tag{9.47}$$

where

$$\Delta\bar{v}^n_i = \text{TVD\_limiter}\Big[\bar{v}^n_{i+1} - \bar{v}^n_i, \bar{v}^n_i - \bar{v}^n_{i-1}\Big]. \tag{9.48}$$

Similar to PLM, the slope limiting can be carried out either in primitive or characteristic variables. For the latter option we can implement projection operators between the two variable spaces. For the m-th component it follows as,

$$\Delta\bar{v}^n_{i:m} = \sum_{k=1}^{3} r^{(k)}_{i:m}\Delta w^{(k)}_i, \tag{9.49}$$

where

$$\Delta w^{(k)}_i = \text{TVD\_limiter}\Big[\mathbf{l}^{(k)}_i \cdot (\bar{\mathbf{V}}^n_{i+1} - \bar{\mathbf{V}}^n_i), \mathbf{l}^{(k)}_i \cdot (\bar{\mathbf{V}}^n_i - \bar{\mathbf{V}}^n_{i-1})\Big]. \tag{9.50}$$

Before proceeding to the second principle (i.e., the characteristic tracing), PPM needs to check if *Condition 1* and *Condition 2* hold in order to make sure the monotonicity property is met in constructing $v^n_{L;R,i}$ based on the first principle of reconstruction.

Condition 1: The monotonic profile of $p_i(x)$ (see Fig. 3) can be ensured by applying the following two constraints:

1. Reducing to the flat FOG reconstruction, $v^n_{L;R,i} = \bar{v}^n_i$, when the two PPM states are newly producing a local extremum on I_i (see panel (a) in Fig. 3). That is, PPM reduces to FOG when

$$(v^n_{R,i} - \bar{v}^n_i)(\bar{v}^n_i - v^n_{L,i}) \le 0. \tag{9.51}$$

2. Recalculate one of the two states by shifting the abscissa of the parabola to the closer interface, either $x_{i-\frac{1}{2}}$ (panel (b) in Fig. 3) or $x_{i+\frac{1}{2}}$ (the bottom right panel in Fig. 3), so that any extremum on I_i is relocated to one of $x_{i\pm1/2}$. PPM further corrects the reconstructed profile at the other interface according to the new monotonic profile. This can be accomplished by checking:

$$\text{(a)} \qquad v^n_{R,i} = 3\bar{v}^n_i - 2v^n_{L,i}, \tag{9.52}$$
$$\text{if } -(v^n_{R,i} - v^n_{L,i})^2 > 6(v^n_{R,i} - v^n_{L,i})\Big(\bar{v}^n_i - (v^n_{R,i} + v^n_{L,i})/2\Big),$$

$$\text{(b)} \qquad v^n_{L,i} = 3\bar{v}^n_i - 2v^n_{R,i}, \tag{9.53}$$
$$\text{if } (v^n_{R,i} - v^n_{L,i})^2 < 6(v^n_{R,i} - v^n_{L,i})\Big(\bar{v}^n_i - (v^n_{R,i} + v^n_{L,i})/2\Big).$$

Condition 2: The two constraints, $v^n_{L,i} \in [\min(\bar{v}^n_{i-1}, \bar{v}^n_i), \max(\bar{v}^n_{i-1}, \bar{v}^n_i)]$ and $v^n_{R,i} \in [\min(\bar{v}^n_i, \bar{v}^n_{i+1}), \max(\bar{v}^n_i, \bar{v}^n_{i+1})]$, are automatically guaranteed by utilizing TVD slope limiters as in Eqs. (9.48) and (9.50). See also the right panel in Fig.

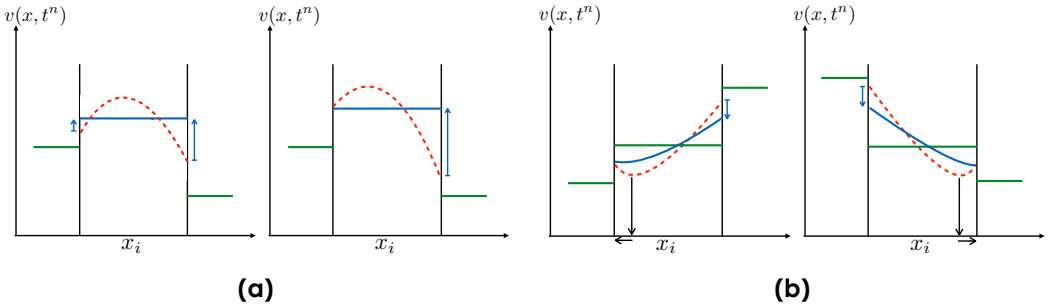


Figure 3. (a) PPM reduces to the flat reconstruction (blue solid line) at the cell I_i on which the original reconstructed parabolic profile (red-dotted line) produces a local extremum, violating the monotonicity constraint in *Condition 1*. In this case, PPM clips at the extremum and becomes FOG. (b) Maintaining monotonic profile by changing the abscissa $x = x_i - \frac{c_1}{2c_2}$ to the closer interface location, either $x_{i-\frac{1}{2}}$ (left panel) or $x_{i+\frac{1}{2}}$ (right panel). See the arrows illustrating the operation. The resulting new monotonic parabola, denoted in blue solid line, produces a new right (or left) state at $x_{i+\frac{1}{2}}$ (or $x_{i-\frac{1}{2}}$) on I_i. In all panels, the green solid lines represent the cell-averaged quantities on each cell.

1.

Remark: As can be seen, the monotonicity constraints in Eq. (9.51) in *Condition 1* and *Condition 2* often become too strong at smooth extrema reducing the truncation error, so-called 'clipping-error', at those locations to first-order. This clipping behavior diminishes the method's formal order of accuracy, considering for smooth solutions away from extrema, from third to first. There have been new approaches for PPM to overcome this drawback.

1.4.2. Step 2: Characteristic Tracing The PPM proceeds to the next step that advances the $v^n_{L;R,i}$ states $v^{n+\frac{1}{2}}_{L;R,i}$ by tracing characteristics. The approach to take is exactly the same as what we have done in PLM, conducting steps in Eqs. (9.24)-(9.28), but with the third-order polynomial defined by Eqs. (9.33)-(9.38) and Eq. (9.46). The integrand for the right state in PPM becomes

$$r^{(k)}_{i:m} \sum_{s=1}^{3} l^{(k)}_{i:s} \Big(c_{0:s} + c_{1:s}(x - x_i) + c_{2:s}(x - x_i)^2 \Big). \tag{9.54}$$

Let us adopt a vector notation $\mathbf{C}_l = (c_{l:1}, \ldots, c_{l:3})^T$, for $l = 0, 1, 2$ in the below, where each $c_{l,m}$ denotes the l-th coefficient of the reconstruction of the m-th primitive variable. See Eq. (9.15). Similar to PLM, we can show that the integrations in Eq. (9.17) of the first two constant and linear terms in Eq.

(9.54) are

$$\sum_k r_{i:m}^{(k)} \mathbf{l}_i^{(k)} \cdot \mathbf{C}_0, \tag{9.55}$$

and

$$\frac{1}{2} \sum_{k;\lambda^{(k)}>0} \Big(1 - \frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big) r_{i:m}^{(k)} \Delta c_1^{(k)}, \tag{9.56}$$

where

$$\Delta c_1^{(k)} = \sum_{s=1}^{3} l_{i:s}^{(k)} c_{1:s} \Delta x = \mathbf{l}_i^{(k)} \cdot \mathbf{C}_1 \Delta x. \tag{9.57}$$

Finally, integrating the last quadratic term in Eq. (9.54) yields

$$\frac{1}{4} \sum_{k;\lambda^{(k)}>0} \left(1 - \frac{2\lambda_i^{(k)} \Delta t}{\Delta x} + \frac{4}{3}\Big(\frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big)^2\right) r_{i:m}^{(k)} \Delta c_2^{(k)}, \tag{9.58}$$

where

$$\Delta c_2^{(k)} = \sum_{s=1}^{3} l_{i:s}^{(k)} c_{2:s} \Delta x^2 = \mathbf{l}_i^{(k)} \cdot \mathbf{C}_2 \Delta x^2. \tag{9.59}$$

Putting all things together, we complete computing the half-time evolved normal predicted states for the third-order PPM in vector form,

$$\begin{aligned} \mathbf{V}_{R,i}^{n+\frac{1}{2}} = \mathbf{C}_0 &+ \frac{1}{2} \sum_{k;\lambda^{(k)}>0} \Big(1 - \frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big) r_i^{(k)} \Delta c_1^{(k)} \\ &+ \frac{1}{4} \sum_{k;\lambda^{(k)}>0} \left(1 - \frac{2\lambda_i^{(k)} \Delta t}{\Delta x} + \frac{4}{3}\Big(\frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big)^2\right) r_i^{(k)} \Delta c_2^{(k)}, \end{aligned} \tag{9.60}$$

and

$$\begin{aligned} \mathbf{V}_{L,i}^{n+\frac{1}{2}} = \mathbf{C}_0 &+ \frac{1}{2} \sum_{k;\lambda^{(k)}<0} \Big(-1 - \frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big) r_i^{(k)} \Delta c_1^{(k)} \\ &+ \frac{1}{4} \sum_{k;\lambda^{(k)}<0} \left(1 + \frac{2\lambda_i^{(k)} \Delta t}{\Delta x} + \frac{4}{3}\Big(\frac{\lambda_i^{(k)} \Delta t}{\Delta x}\Big)^2\right) r_i^{(k)} \Delta c_2^{(k)}. \end{aligned} \tag{9.61}$$

Obviously, we notice that the temporally evolved PPM states are very similar to those of PLM in Eqs. (9.31) and (9.32) up to the first two terms. The last terms in Eqs. (9.60) and (9.61) appear as PPM's additional terms from the quadratic term in Eq. (9.33).

1.5. The Fifth-order WENO Method

We study two formulations of the fifth-order weighted essentially non-oscillatory (WENO) scheme as another choice of the high-order methods. The first is the classical fifth-order WENO scheme by Jiang and Shu (1996), denoted with WENO5. The second approach, referred to as WENO-Z by Borges et al. (2008), is a variant of WENO5, with an improved formulation of weights. The two schemes follow most of the formulations in the same way, only differing in implementing the nonlinear weights.

Our WENO implementations adopt the following procedures:

- Use either WENO5 or WENO-Z to reconstruct $v_{L;R,i}^{weno,n}$.
- Evolve these WENO reconstructed profiles by $\Delta t/2$ to get $v_{L;R,i}^{weno,n+\frac{1}{2}}$ via the same characteristic tracing approach in PPM.

In practice, we perform the following steps:

1. Take the same third-order polynomial $p_i(x)$ in PPM defined by Eqs. (9.33) - (9.38), replacing the PPM's fourth-order reconstructed states $v_{L;R,i}^{ppm,n}$ in Eq. (9.46) by the fifth-order WENO states $v_{L;R,i}^{weno,n}$. This means that we skip all the steps in PPM that use $\phi_{\pm}(x)$ in Eqs. (9.39) - (9.46).
2. Check only the second constraint in *Condition 1* which is related to preserving monotonicity of the parabolic profile $p_i(x)$ on each I_i *with* the left and right states $v_{L;R,i}^{weno,n}$. Notice that the rest of the constraints are not needed as WENO provides non-oscillatory states by design, which we describe in the below.
3. Conduct the steps for characteristic tracing in Eqs. (9.54) - (9.61).

In general, WENO is best formulated with one of the high-order ODE solvers such as Runge-Kutta discretization schemes. In this way, one can establish an expected overall high-order accuracy in both spatial and temporal updates. For computational simplicity though, we could integrate WENO methods within the framework of second-order temporal ODE solver following the characteristic tracing formulation that provides second-order overall accuracy in smooth flows.

The main idea in WENO is to adapt nonlinearity in its reconstruction procedure according to smoothness measurements on each of the three ENO stencils, S_l with $l = 1, 2, 3$, each of which consisting three cells I_i, $i = i_1, i_2, i_3$. So let us first define

$$S_1 = \{I_{i-2}, I_{i-1}, I_i\}, \tag{9.62}$$
$$S_2 = \{I_{i-1}, I_i, I_{i+1}\}, \tag{9.63}$$
$$S_3 = \{I_i, I_{i+1}, I_{i+2}\}. \tag{9.64}$$

The two WENO reconstructions consist of the following three steps:

1.5.1. Step 1: ENO-Build : We begin with building three second-degree ENO polynomials for each $l = 1, 2, 3$,

$$p_l(x) = \sum_{k=0}^{2} a_{l,k}(x - x_i)^k, \tag{9.65}$$

each of which is defined on S_l, satisfying

$$\frac{1}{\Delta x}\int_{I_k} p_l(x)\,dx = \bar{v}_k, \tag{9.66}$$

for $k = i + l - 3, \ldots, i + l - 1$. After a bit of algebra, we obtain the coefficients $a_{l,k}$ that determine $p_l(x)$ in Eq. (9.65).

For $l = 1$,

$$a_{1,0} = \left(-\frac{1}{24}\bar{v}_{i-2} + \frac{1}{12}\bar{v}_{i-1} + \frac{23}{24}\bar{v}_i\right), \tag{9.67}$$

$$a_{1,1} = \left(\frac{1}{2}\bar{v}_{i-2} - 2\bar{v}_{i-1} + \frac{3}{2}\bar{v}_i\right)\frac{1}{\Delta x}, \tag{9.68}$$

$$a_{1,2} = \left(\frac{1}{2}\bar{v}_{i-2} - \bar{v}_{i-1} + \frac{1}{2}\bar{v}_i\right)\frac{1}{\Delta x^2}, \tag{9.69}$$

and for $l = 2$,

$$a_{2,0} = \left(-\frac{1}{24}\bar{v}_{i-1} + \frac{13}{12}\bar{v}_i - \frac{1}{24}\bar{v}_{i+1}\right), \tag{9.70}$$

$$a_{2,1} = \left(-\frac{1}{2}\bar{v}_{i-1} + \frac{1}{2}\bar{v}_{i+1}\right)\frac{1}{\Delta x}, \tag{9.71}$$

$$a_{2,2} = \left(\frac{1}{2}\bar{v}_{i-1} - \bar{v}_i + \frac{1}{2}\bar{v}_{i+1}\right)\frac{1}{\Delta x^2}. \tag{9.72}$$

Lastly, for $l = 3$, we get

$$a_{3,0} = \left(\frac{23}{24}\bar{v}_i + \frac{1}{12}\bar{v}_{i+1} - \frac{1}{24}\bar{v}_{i+2}\right), \tag{9.73}$$

$$a_{3,1} = \left(-\frac{3}{2}\bar{v}_i + 2\bar{v}_{i+1} - \frac{1}{2}\bar{v}_{i+2}\right)\frac{1}{\Delta x}, \tag{9.74}$$

$$a_{3,2} = \left(\frac{1}{2}\bar{v}_i - \bar{v}_{i+1} + \frac{1}{2}\bar{v}_{i+2}\right)\frac{1}{\Delta x^2}. \tag{9.75}$$

Then three sets of left and right states follow as

$$\{p_1(x_{i-\frac{1}{2}}), p_2(x_{i-\frac{1}{2}}), p_3(x_{i-\frac{1}{2}})\},\ \text{and}\ \{p_1(x_{i+\frac{1}{2}}), p_2(x_{i+\frac{1}{2}}), p_3(x_{i+\frac{1}{2}})\}, \tag{9.76}$$

where each of $p_l(x_{i\pm 1/2})$ is the ENO approximation and given by, first for p_1,

$$p_1(x_{i-\frac{1}{2}}) = -\frac{1}{6}\bar{v}_{i-2} + \frac{5}{6}\bar{v}_{i-1} + \frac{1}{3}\bar{v}_i, \tag{9.77}$$

$$p_1(x_{i+\frac{1}{2}}) = \frac{1}{3}\bar{v}_{i-2} - \frac{7}{6}\bar{v}_{i-1} + \frac{11}{6}\bar{v}_i, \tag{9.78}$$

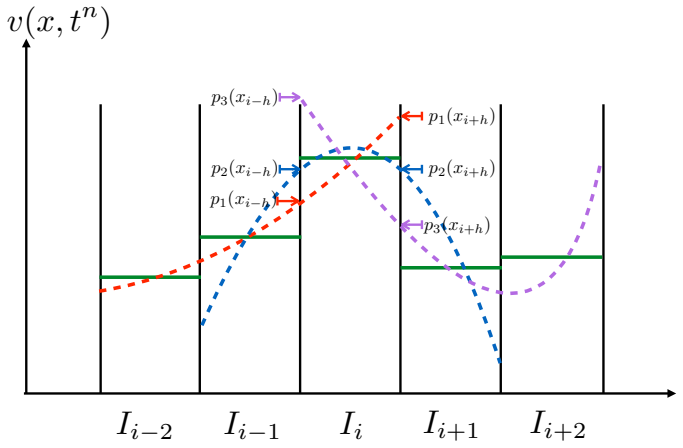


Figure 4. In the plot the shorthand index 'h' represents the half-index $\frac{1}{2}$. WENO reconstruction using three ENO approximations $p_l(x)$, $l = 1, 2, 3$ on $S = \cup_l S_l$. The dotted lines in red, blue and purple respectively illustrate $p_1(x)$ on S_1, $p_2(x)$ on S_2, and $p_3(x)$ on S_3, each of which preserves cell-volume quantities $\bar{v}_k$ (represented in green lines) on each I_k. The nodal values $p_l(x_{i\pm1/2})$ are marked at the cell interfaces. They are combined with the nonlinear weights $\omega_l^{\pm}$ to compute the fifth-order accurate WENO states $v_{L;R,i}^{weno,n} = \sum_{l=1}^{3} \omega_l^{\pm} p_l(x_{i\pm1/2})$.

and for p_2,

$$p_2(x_{i-\frac{1}{2}}) = \frac{1}{3}\bar{v}_{i-1} + \frac{5}{6}\bar{v}_i - \frac{1}{6}\bar{v}_{i+1}, \tag{9.79}$$

$$p_2(x_{i+\frac{1}{2}}) = -\frac{1}{6}\bar{v}_{i-1} + \frac{5}{6}\bar{v}_i + \frac{1}{3}\bar{v}_{i+1}, \tag{9.80}$$

and finally for p_3,

$$p_3(x_{i-\frac{1}{2}}) = \frac{11}{6}\bar{v}_i - \frac{7}{6}\bar{v}_{i+1} + \frac{1}{3}\bar{v}_{i+2}, \tag{9.81}$$

$$p_3(x_{i+\frac{1}{2}}) = \frac{1}{3}\bar{v}_i + \frac{5}{6}\bar{v}_{i+1} - \frac{1}{6}\bar{v}_{i+2}. \tag{9.82}$$

These left and right states respectively approximate the nodal value $v(x_{x_{i\pm1/2}})$ with third-order accuracy, i.e., $p_l(x_{i\pm1/2}) - v(x_{i\pm1/2}) = O(\Delta x^3)$, by using the given cell-averaged quantities $\bar{v}_k$.

1.5.2. Step 2: Linear Constant Weights The next step is to construct a fourth-degree polynomial

$$\phi(x) = \sum_{k=0}^{4} b_k (x - x_i)^k \tag{9.83}$$

over the entire stencil $S = \cup_{l=1}^{3} S_l$, also satisfying

$$\frac{1}{\Delta x}\int_{I_k} \phi(x)dx = \bar{v}_k, \tag{9.84}$$

for $k = i-2, \ldots, i+2$. We can show that the coefficients b_k are given as

$$b_0 = \frac{3}{640}\bar{v}_{i-2} - \frac{29}{480}\bar{v}_{i-1} + \frac{1067}{960}\bar{v}_i - \frac{29}{480}\bar{v}_{i+1} + \frac{3}{640}\bar{v}_{i+2}, \tag{9.85}$$

$$b_1 = \frac{5}{48}\bar{v}_{i-2} - \frac{17}{24}\bar{v}_{i-1} + \frac{17}{24}\bar{v}_{i+1} - \frac{5}{48}\bar{v}_{i+2}, \tag{9.86}$$

$$b_2 = -\frac{1}{16}\bar{v}_{i-2} + \frac{3}{4}\bar{v}_{i-1} - \frac{11}{8}\bar{v}_i + \frac{3}{4}\bar{v}_{i+1} - \frac{1}{16}\bar{v}_{i+2}, \tag{9.87}$$

$$b_3 = -\frac{1}{12}\bar{v}_{i-2} + \frac{1}{6}\bar{v}_{i-1} - \frac{1}{6}\bar{v}_{i+1} + \frac{1}{12}\bar{v}_{i+2}, \tag{9.88}$$

$$b_4 = \frac{1}{24}\bar{v}_{i-2} - \frac{1}{6}\bar{v}_{i-1} + \frac{1}{4}\bar{v}_i - \frac{1}{6}\bar{v}_{i+1} + \frac{1}{24}\bar{v}_{i+2}. \tag{9.89}$$

We use $\phi(x)$ to determine three linear constant weights $\gamma_l^{\pm}$, $l = 1, 2, 3$, with $\sum_l \gamma_l^{\pm} = 1$, such that

$$\phi(x_{i\pm 1/2}) = \sum_{l=1}^{3} \gamma_l^{\pm} p_l(x_{i\pm 1/2}). \tag{9.90}$$

The values on the left-hand side become

$$\phi(x_{i-\frac{1}{2}}) = -\frac{1}{20}\bar{v}_{i-2} + \frac{9}{20}\bar{v}_{i-1} + \frac{47}{60}\bar{v}_i - \frac{13}{60}\bar{v}_{i+1} + \frac{1}{30}\bar{v}_{i+2}, \tag{9.91}$$

and

$$\phi(x_{i+\frac{1}{2}}) = \frac{1}{30}\bar{v}_{i-2} - \frac{13}{60}\bar{v}_{i-1} + \frac{47}{60}\bar{v}_i + \frac{9}{20}\bar{v}_{i+1} - \frac{1}{20}\bar{v}_{i+2}. \tag{9.92}$$

Now, by inspection, one gets for the left state,

$$\gamma_1^- = \frac{3}{10}, \gamma_2^- = \frac{6}{10}, \gamma_3^- = \frac{1}{10}, \tag{9.93}$$

and for the right state,

$$\gamma_1^+ = \frac{1}{10}, \gamma_2^+ = \frac{6}{10}, \gamma_3^+ = \frac{3}{10}. \tag{9.94}$$

1.5.3. Step 3: Nonlinear Weights The last step that imposes the non-oscillatory feature in the WENO approximations is to measure how smooth the three polynomials $p_l(x)$ vary on I_i. This is done by determining non-constant, nonlinear weights $\omega_l^{\pm}$ (three of them for each $\pm$ state) that rely on the so-called smoothness indicator β_l.

In most of the WENO papers, the smoothness indicator takes of the form of a scaled sum of the square L_2-norms of all the derivatives up to the degree of $p_l(x)$ on I_i. This is to say, in our case of deg $p_l(x) = 2$ for all $l = 1, 2, 3$,

$$\beta_l = \sum_{s=1}^{2} \left(\Delta x^{2s-1} \int_{I_i} \Big[\frac{d^s}{dx^s} p_l(x) \Big]^2 dx \right), \tag{9.95}$$

where the scaling factor Δx^{2s-1} removes the grid size Δx dependency in measuring the norm. With this definition, β_l becomes small for smooth flows, and large for discontinuous flows.

For explicit expressions, we attain

$$\beta_1 = \frac{13}{12} \left(\bar{v}_{i-2} - 2\bar{v}_{i-1} + \bar{v}_i \right)^2 + \frac{1}{4} \left(\bar{v}_{i-2} - 4\bar{v}_{i-1} + 3\bar{v}_i \right)^2, \tag{9.96}$$

$$\beta_2 = \frac{13}{12} \left(\bar{v}_{i-1} - 2\bar{v}_i + \bar{v}_{i+1} \right)^2 + \frac{1}{4} \left(\bar{v}_{i-1} - \bar{v}_{i+1} \right)^2, \tag{9.97}$$

$$\beta_3 = \frac{13}{12} \left(\bar{v}_i - 2\bar{v}_{i+1} + \bar{v}_{i+2} \right)^2 + \frac{1}{4} \left(3\bar{v}_i - 4\bar{v}_{i+1} + \bar{v}_{i+2} \right)^2. \tag{9.98}$$

Equipped with these β_l, the nonlinear weights $\omega_l^{\pm} \geq 0$ are defined as: (i) for WENO5,

$$\omega_l^{\pm} = \frac{\tilde{\omega}_l^{\pm}}{\sum_s \tilde{\omega}_s^{\pm}}, \text{ where } \tilde{\omega}_l^{\pm} = \frac{\gamma_l^{\pm}}{(\epsilon + \beta_l)^m}, \tag{9.99}$$

and (ii) for WENO-Z,

$$\omega_l^{\pm} = \frac{\tilde{\omega}_l^{\pm}}{\sum_s \tilde{\omega}_s^{\pm}}, \text{ where } \tilde{\omega}_l^{\pm} = \gamma_l^{\pm} \left(1 + \Big(\frac{|\beta_0 - \beta_2|}{\epsilon + \beta_l} \Big)^m \right). \tag{9.100}$$

Here ϵ is any arbitrarily small positive number that prevents division by zero, for which we choose $\epsilon = 10^{-36}$. The WENO reconstruction is scale invariant as long as ϵ is chosen to be a small percentage of the size of typical v_i under consideration.

It should be noted that the nonlinear weights, by design, satisfy the following two requirements:

1. The nonlinear weights become equivalent to the linear ones, $\omega_l^{\pm} \approx \gamma_l^{\pm}$, when the quantity $v(x)$ WENO approximates is smooth over the entire stencil S.

2. Otherwise, $\omega_j^{\pm} \approx 0$ if $v(x)$ is discontinuous on one of $S = \cup_l S_l$, say, S_j, but is smooth on at least one of $\cup_{l \neq j} S_l$.

Most of the WENO literatures use $m = 2$ for the power in the denominator in Eq. (9.99), which determines the rate of changes in β_l. However, we observe that using $m = 1$ resolves discontinuities sharper in most of our numerical simulations, so the default value in our implementation.

Using these nonlinear weights, we complete the WENO reconstruction procedure with the fifth-oder spatially accurate reconstructed values,

$$v_{L;R,i}^{weno,n} = \sum_{l=1}^{3} \omega_l^{\pm} p_l(x_{i\pm 1/2}) \tag{9.101}$$

The remaining tasks are to conduct the steps for characteristic tracing described in Eqs. (9.54) - (9.61), which produce the Riemann states $(v_L, v_R) = (v_{R,i+\frac{1}{2}}^{weno,n+\frac{1}{2}}, v_{L,i+1}^{weno,n+\frac{1}{2}})$. They are provided as the initial value problems for the Godunov fluxes at each interface $x_{i+\frac{1}{2}}$.

2. 1D Shock Tube Results and Method Comparsion

The Shu-Osher problem (1989) tests a shock-capturing scheme's ability to resolve small-scale flow features. It gives a good indication of the numerical (artificial) viscosity of a method. Since it is designed to test shock-capturing schemes, the equations of interest are the one-dimensional Euler equations for a single-species perfect gas.

In this problem, a (nominally) Mach 3 shock wave propagates into a sinusoidal density field. As the shock advances, two sets of density features appear behind the shock. One set has the same spatial frequency as the un-shocked perturbations, but for the second set, the frequency is doubled. Furthermore, the second set follows more closely behind the shock. None of these features is spurious. The test of the numerical method is to accurately resolve the dynamics and strengths of the oscillations behind the shock.

The problem is initialized as follows. On the domain $-4.5 \le x \le 4.5$, the shock is at $x = x_s$ at $t = 0.0$. On either side of the shock,

$$\mathbf{V}(x,0) = \begin{cases} \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_L = \begin{pmatrix} 3.857143 \\ 2.629369 \\ 10.33333 \end{pmatrix} & \text{if } x \le x_s, \\ \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}_R = \begin{pmatrix} 1 + a_\rho \sin(f_\rho x) \\ 0.0 \\ 1.0 \end{pmatrix} & \text{if } x > x_s. \end{cases} \tag{9.102}$$

where a_ρ is the amplitude and f_ρ is the frequency of the density perturbations, for which we take $a_\rho = 0.2$ and $f_\rho = 5.0$. The ideal equation of state is used with γ set to 1.4. The location of the initial discontinuity is at $x_s = -4.0$.

For this problem, special boundary conditions are applied. The initial conditions should not change at the boundaries; if they do, errors at the boundaries can contaminate the results. To avoid this possibility, a boundary condition subroutine was written to set the boundary values to their initial values.

The purpose of the tests is to compare computed solutions using five different reconstruction methods of first-order Godunov, PLM, PPM, WENO-5 and WENO-Z. Therefore, all computations are carried out on a uniform mesh resolution of $N = 200$. Solutions in Fig. 2. are obtained at $t = 1.8$. The reference solution, using 4000 mesh cells, is overplotted in black curve with five different computed solutions in Fig. 2. This solution was computed using PLM at a CFL number of $C_a = 0.8$.

It is evident that the higher-order methods such as PPM, WENO-5 and WENO-Z resolve better resolutions producing much higher peaks and troughs in the oscillating density shapes. The first-order Godunov solution is the most dissipative among all, essentially failing to resolve sufficiently the high frequency oscillating regions at all. The last panel clearly indicate the great advantage of using high-order methods over the low-order methods on a given size of grid resolution.

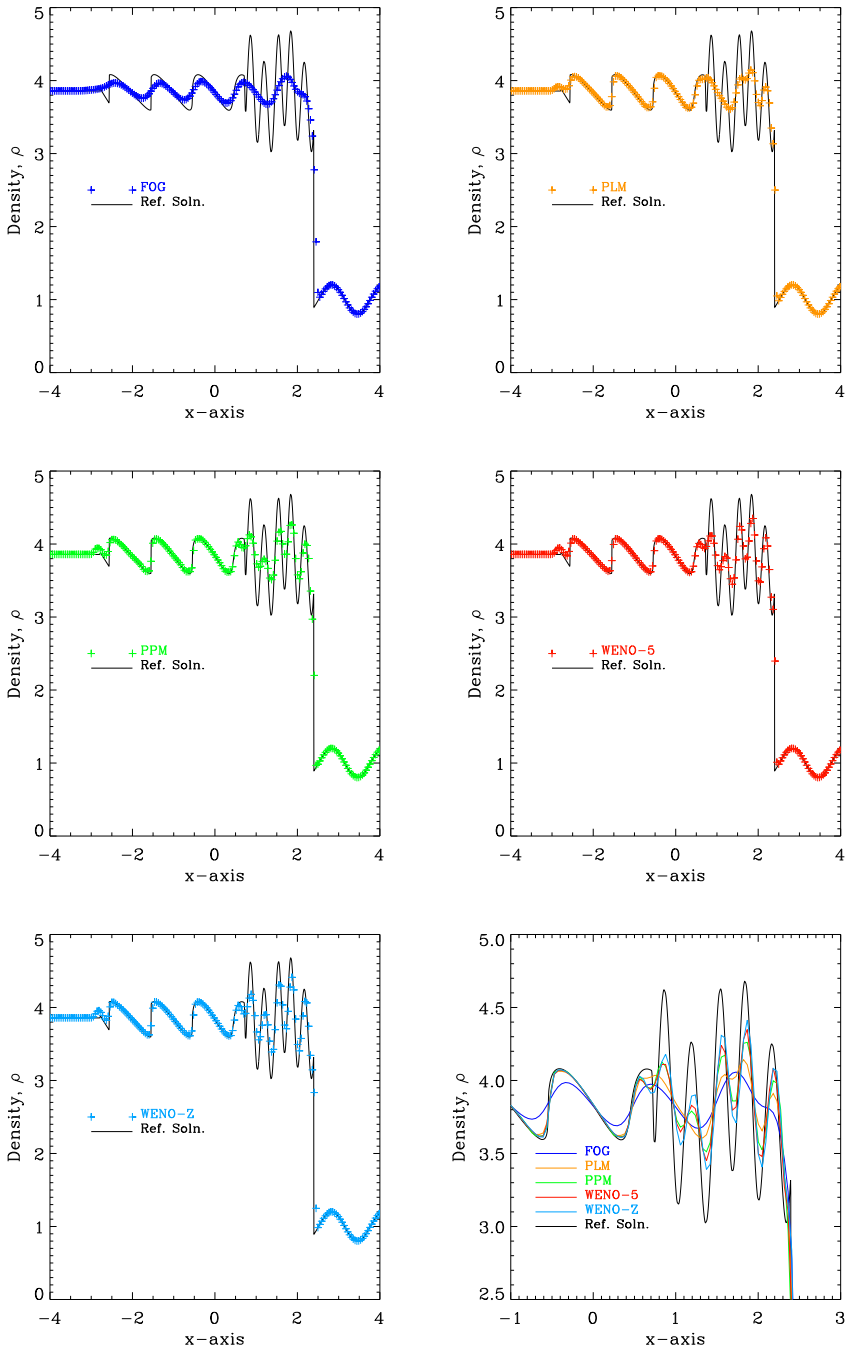


Figure 5. The Shu-Osher problem. The density profiles at $t = 1.8$ are computed using five different reconstruction schemes of first-order Godunov (FOG), PLM, PPM, WENO-5 and WENO-Z on a 200 grid resolution. The reference solution is obtained using PLM on a 4000 grid resolution. The last panel illustrates a closeup view of the five different numerical solutions in the domain between $-1 \leq x \leq 3$.

Chapter 10

Multidimensional Euler Equations

In this chapter we are concerned with multidimensional hyperbolic system of conservation laws. For Cartesian geometry we can write the equations of our interest as

$$\mathbf{U}_t + \nabla \cdot \mathbf{Flux}(\mathbf{U}) = \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y + \mathbf{H}(\mathbf{U})_z = \mathbf{0}, , \tag{10.1}$$

where we take the conventional notation for multidimensional fluxes $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ for x, y and z directions, respectively.

For exposition purposes, we shall present one of the two ways of solving Eq. 10.1 in 2D. The approach we will take is the simpler one of the two, called *dimensionally split methods*. The other approach, which is in general more computationally expensive but more accurate, is called *dimensionally unsplit methods*, for which we simply provide a list of references.

1. Two-Dimensional Euler Equations in Conservative Form

We write the full Euler equations in 2D in the conservative form as

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{0}, \tag{10.2}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ u \\ v \\ w \\ E \end{pmatrix}, \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{pmatrix}, \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{pmatrix}, \tag{10.3}$$

Since $\frac{\partial}{\partial z} = 0$ the z-velocity component w becomes simply an passively advected quantity in 2D.

2. Dimensionally Split Methods

For nonlinear systems dimensional splitting is not exact nor the best way to use, but one may construct an approximate splitting scheme very easily, especially when extended from an already existing 1D code. Consider the 2D initial value problem

$$\begin{cases} \text{PDE:} & \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{0}, \\ \text{IC:} & \mathbf{U}(x, y, t^n) = \mathbf{U}^n. \end{cases} \tag{10.4}$$

The two dimensional splitting approach replaces Eq. 10.4 by a pair of one dimensional IVPs

$$\left\{ \begin{array}{ll} \text{PDE:} & \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = \mathbf{0} \\ \text{IC:} & \mathbf{U}^n \end{array} \right\} \overset{\Delta t}{\Longrightarrow} \mathbf{U}^{n+\frac{1}{2}}, \tag{10.5}$$

and

$$\left\{ \begin{array}{ll} \text{PDE:} & \mathbf{U}_t + \mathbf{G}(\mathbf{U})_y = \mathbf{0} \\ \text{IC:} & \mathbf{U}^{n+\frac{1}{2}} \end{array} \right\} \overset{\Delta t}{\Longrightarrow} \mathbf{U}^{n+1}, \tag{10.6}$$

where Δt is chosen to satisfy the multidimensional CFL condition

$$\max_{x,y} \left\{ \frac{|\lambda_x|}{\Delta x}, \frac{|\lambda_y|}{\Delta y} \right\} \Delta t \leq 1. \tag{10.7}$$

Here the maximum wave speed calculation takes the global maximum value over the all available wave speeds in x and y wave characteristics evaluated over an entire computational domain. In practice for the Euler equations,

$$\max_{x} \left\{ \frac{|\lambda_x|}{\Delta x}, \frac{|\lambda_y|}{\Delta y} \right\} = \max_{i,j} \left\{ \frac{|u_{i,j}| + c_{si,j}}{\Delta x}, \frac{|v_{i,j}| + c_{si,j}}{\Delta y} \right\} \tag{10.8}$$

As we can see in the first step in Eq. 10.5 we solve a 1D problem in the x-direction for a time step Δt. This is called the *x-sweep* and its solution only reflects a *half-updated state* $\mathbf{U}^{n+1/2}$ from the x-directional flux contribution. In the next step in Eq. 10.6 we solve another 1D problem in the y-direction, also for the same time step Δt. This is called the *y-sweep* which takes the half-updated state $\mathbf{U}^{n+1/2}$ as an initial condition.

Let $\mathcal{X}^{(t)}$ and $\mathcal{Y}^{(t)}$ be the operators to approximate the solutions in Eq. 10.5 and Eq. 10.6, respectively. Then we can express Eq. 10.5 and Eq. 10.6 as either

$$\mathbf{U}^{n+1} = \mathcal{Y}^{(\Delta t)} \mathcal{X}^{(\Delta t)} \mathbf{U}^n, \tag{10.9}$$

or

$$\mathbf{U}^{n+1} = \mathcal{X}^{(\Delta t)} \mathcal{Y}^{(\Delta t)} \mathbf{U}^n, \tag{10.10}$$

since there is no particular reason for applying the operators in any specific order.

These splitting approaches in Eqs. (10.9) - (10.10) can be shown to be only first-order in time (Strang, 1968) if the individual operators $\mathcal{X}$ and $\mathcal{Y}$ are at least first-order accurate in time. Alternatively, one can obtain a more efficient second-order accurate splitting (Strang, 1968),

$$\mathbf{U}^{n+1} = \mathcal{X}^{(\frac{\Delta t}{2})}\mathcal{Y}^{(\Delta t)}\mathcal{X}^{(\frac{\Delta t}{2})}\mathbf{U}^{n}, \tag{10.11}$$

or

$$\mathbf{U}^{n+1} = \mathcal{Y}^{(\frac{\Delta t}{2})}\mathcal{X}^{(\Delta t)}\mathcal{Y}^{(\frac{\Delta t}{2})}\mathbf{U}^{n}, \tag{10.12}$$

which require only 50% more work than the first splitting approaches in Eqs. (10.9) - (10.10).

Yet another type of second-order accurate scheme is

$$\mathbf{U}^{n+2} = \mathcal{X}^{(\Delta t)}\mathcal{Y}^{(\Delta t)}\mathcal{Y}^{(\Delta t)}\mathcal{X}^{(\Delta t)}\mathbf{U}^{n}, \tag{10.13}$$

or

$$\mathbf{U}^{n+2} = \mathcal{Y}^{(\Delta t)}\mathcal{X}^{(\Delta t)}\mathcal{X}^{(\Delta t)}\mathcal{Y}^{(\Delta t)}\mathbf{U}^{n}, \tag{10.14}$$

which is second-order accurate every other time step and has been implemented and used in the FLASH's split PPM hydrodynamics solver.

3. Dimensionally Unsplit Methods

Alternative to splitting multidimensional PDEs into sub-1D systems as in the dimensionally splitting methods, we can directly discretize and numerically solve the whole multidimensional PDEs. This is called the *directionally unsplit methods*. With this unsplit approach one can avoid introducing the numerical errors from splitting PDEs dimensionally.

The unsplit methods are in general better in maintaining multidimensional symmetries than the split methods. Other challenges of the unsplit approaches include that (i) one has to require more memory spaces to store all the available calculations in intermediate steps, (ii) the unsplit consideration is more attended to account for multidimensional wave structures, (iii) a more careful multidimensional stability needs to be established in order to use the full CFL stability region, i.e., $0 \leq C_a \leq 1$. Otherwise, the CFL region will be reduced to $0 \leq C_a \leq 1/N_{dim}$ in general.

For those who are interested in more reading, please take a look at the following references:

- Multidimensional Upwind Methods for Hyperbolic Conservation Laws by Colella, 1990, JCP (attached; 2D hydrodynamics),
- An Unsplit Staggered Mesh Scheme for Multidimensional Magnetohydrodynamics, D. Lee and A. Deane, 2009, JCP (2D MHD),
- An Unsplit 3D Upwind Method for Hyperbolic Conservation Laws, Saltzman, 1994, JCP (3D hydrodynamics),

- A Solution Accurate, Efficient and Stable Unsplit Staggered Mesh Scheme for Three Dimensional Magnetohydrodynamics, D. Lee, 2013, JCP (3D MHD),
- Riemann Solvers and Numerical Methods for Fluid Dynamics, Toro, Springer,
- Finite-Volume Methods for Hyperbolic Problems, LeVeque, Cambridge Texts in Applied Mathematics,
- And many others!

JOURNAL OF COMPUTATIONAL PHYSICS 87, 171–200 (1990)

Multidimensional Upwind Methods for Hyperbolic Conservation Laws*

PHILLIP COLELLA

Mechanical Engineering Department, University of California, Berkeley, California 94720

Received June 20, 1984; revised October 21, 1987

We present a class of second-order conservative finite difference algorithms for solving numerically time-dependent problems for hyperbolic conservation laws in several space variables. These methods are upwind and multidimensional, in that the numerical fluxes are obtained by solving the characteristic form of the full multidimensional equations at the zone edge, and that all fluxes are evaluated and differenced at the same time; in particular, operator splitting is not used. Correct behavior at discontinuities is obtained by the use of solutions to the Riemann problem, and by limiting some of the second-order terms. Numerical results are presented, which show that the methods described here yield the same high resolution as the corresponding operator split methods.

INTRODUCTION

Over the last several years, there has been considerable development of upwind-type numerical methods for solving nonlinear systems of hyperbolic conservation laws in several space dimensions. These methods, generally speaking, are all second-order extensions of Godunov's first-order method [11]. They incorporate into the numerical solutions the nonlinear wave propagation properties of the solution, in the form of Riemann problems and characteristic equations, leading to algorithms which are robust and accurate, even in the presence of nonlinear discontinuities. However, all of the methods currently in use are derived using the characteristic form of the equations in one space dimension, with most of these algorithms being extented to several space dimensions using operator splitting. Nonetheless, these algorithms, particularly the operator split ones, have been quite successful in resolving complex patterns of interacting discontinuities and smooth waves; for further details see [22] and the references cited there.

In this paper, we will consider a class of conservative finite difference algorithms

* Work supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research of the U.S. Department of Energy at the Lawrence Berkeley Laboratory under Contract DE-AC03-76SF00098; by the U.S. Defence Nuclear Agency under DNA task code Y99QAXSG; and by the Office of Naval Research under Contract N00014-76-C-0316.

0021-9991/90 $3.00

581/87/1-12

for hyperbolic conservation laws in several space variables which do not make use of operator splitting, for which the multidimensional wave propagation properties of the solution are used to calculate fluxes. Unsplit schemes are customarily used in a variety of applications, including petroleum reservoir simulation [18], ionospheric physics [24], and Lagrangian hydrodynamics [1]. Thus, one of our goals is to provide algorithms which have the same robustness and resolution as the existing operator split algorithms, but which have the same unsplit structure as the existing algorithms used in the applications codes in those areas. In addition, there are two specific applications for which these methods were developed which are the subject of our current research. One is as a method to be coupled with a front tracking method [3], where the tracked front is represented locally by a polygonal line which divides the cells into two pieces. In each piece, the solution is updated by a method that is necessarily unsplit, in order to preserve the Rankine–Hugoniot relations for the tracked front. The second application is as a starting point for the extension to more than one space dimension of implicit/explicit methods of the type discussed in [10]. In these methods, propagation along each of the characteristic families is treated implicitly or explicitly, depending on whether the CFL number for that characteristic is greater than or less than 1. Thus we require an explicit algorithm with properties similar to those of the 1-dimensional algorithms in [7], but which can be hybridized continuously to an implicit algorithm, in order to have steady states which are independent of Δt.

The design of the algorithm described here is broken into two steps. First, we specify an algorithm for a linear scalar advection equation, which, in smooth regions, is second-order accurate, to which a monotonicity condition, related to those used in [20] for advection algorithms in one dimension, is applied. We then construct the algorithm for systems by introducing a predictor-corrector formalism and by replacing various derivatives in the predictor step by finite differences, using the advection algorithm as guide: upwind differences for advection become differences of Godunov fluxes for systems, and monotonized central differences for advection become monotonized central differences with monotonicity constraints applied to the appropriate choice of transformed variables. Independently of the present work, van Leer also derived multidimensional upwind methods for hyperbolic conservation laws, following a similar line or reasoning; in particular, both methods lead to the algorithm for advection given in the next section. However, his extension to systems is rather different from the predictor-corrector formalism described here; for details, see [21].

A major problem in the program outlined above is the specification of design criteria which guarantee oscillation-free results, even in the one for a linear scalar equation. The principal criterion in one space dimension is that the scheme be total variation diminishing [13]; however, a straightforward generalization of this criterion to more than one dimension has been shown in [12] to imply that the scheme is at most first-order accurate for smooth solutions. The approach taken in the present work is to specify cetain necessary conditions that the scheme must satisfy, and which are satisfied by the schemes described here. These are:

(1) For a 1-dimensional problem aligned with one of the grid directions, the algorithm should reduce to a second-order Godunov method of a type described in [7].

(2) The second-order scheme without limiting, and the first-order scheme obtained by imposing the full limiting of the fluxes at all mesh points, should have as linear difference schemes, the same CFL stability limit on the time step. This CFL stability limit should be the same as for an operator split scheme, with the component 1-dimensional algorithm as in [7].

(3) In the case of linear advection, the fully limited scheme should satisfy a maximum principle.

In the following, we will restrict our attention to the case of two space variables. Although the formalism developed here carries over to higher dimensions, the trade-offs between performance and cost change as the number of dimensions grow; a proper evaluation of what those trade-offs are can only be made by numerical experimentation. In three dimensions, such a study would strain the capabilities of present computer technology. Some discussion of these considerations will be made in the final section of this paper.

1. Advection Algorithms

We consider the scalar advection equation in two space variables

$$\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \nabla \rho = 0 \tag{1.1}$$

$$\mathbf{x} = (x, y), \qquad \rho = \rho(\mathbf{x}, t) \qquad \nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right) \qquad \mathbf{u} = (u, v) \qquad u, v > 0.$$

We want to solve numerically initial value problems for (1.1). To this end, we will attempt to construct algorithms which generalize upstream-centered algorithms in [20] to two space variables, without replacing the operator approximating the time evolution of (1.1) by the product of 1-dimensional evolution operators. Our strategy will be to start from a well-behaved first-order upwind algorithm for solving (1.1). We add to the evolution operator the terms necessary to make the algorithm second-order accurate in a way such that they can be limited, i.e., subtracted off, at discontinuities.

Let Δx, Δy be spatial increments, Δt a time increment. We assume that we know ρ_{ij}^n, the average of ρ at time t^n:

$$\rho_{i,j}^n = \frac{1}{\sigma_{i,j}} \int_{\Delta_{i,j}} \rho(\mathbf{x}, t^n)\, d\mathbf{x}.$$

Here $\Delta_{i,j} = [(i - \frac{1}{2})\Delta x, (i + \frac{1}{2})\Delta x] \times [(j - \frac{1}{2})\Delta y, (j + \frac{1}{2})\Delta y]$, $\sigma_{i,j} = (\text{area of } \Delta_{i,j})$.

We wish to calculate $\rho_{i,j}^{n+1}$, the solution to (1.1) at time $t^{n+1} = t^n + \Delta t$. A natural algorithm for doing this is to trace backward in time from $t^n + \Delta t$ the set $\Delta_{i,j}$, along the characteristics of (1.1), to obtain $\Delta'_{i,j}$. Then $\rho_{i,j}^{n+1}$ is set equal to the average over $\Delta'_{i,j}$ of the trivial interpolation function $\rho^I(\mathbf{x}) = \rho_{i,j}^n$ if $\mathbf{x} \in \Delta_{i,j}$:

$$\begin{aligned}\rho_{i,j}^{n+1} &= \frac{1}{\sigma'_{i,j}} \int_{\Delta'_{i,j}} \rho^I(x, y)\, dx\, dy \\ &= (A_1 \rho_{i,j}^n + A_2 \rho_{i,j-1}^n + A_3 \rho_{i-1,j}^n + A_4 \rho_{i-1,j-1}^n) \frac{1}{\sigma'_{i,j}} \end{aligned} \tag{1.2}$$

where the A_k's are the areas in each of the four upstream zones swept out by $\mathbf{u}$, as indicated in Fig. 1.

We can put this scheme in explicit conservation form

$$\rho_{i,j}^{n+1} = \rho_{i,j}^n + \frac{u\,\Delta t}{\Delta x}(\rho_{i-1/2,j}^{n+1/2} - \rho_{i+1/2,j}^{n+1/2}) + \frac{v\,\Delta t}{\Delta y}(\rho_{i,j-1/2}^{n+1/2} - \rho_{i,j+1/2}^{n+1/2}) \tag{1.3}$$

$$\begin{aligned}\rho_{i,j+1/2}^{n+1/2} &= \rho_{i,j}^n + \frac{u\,\Delta t}{2\,\Delta x}(\rho_{i-1,j}^n - \rho_{i,j}^n) \\ \rho_{i+1/2,j}^{n+1/2} &= \rho_{i,j}^n + \frac{v\,\Delta t}{2\,\Delta y}(\rho_{i,j-1}^n - \rho_{i,j}^n). \end{aligned} \tag{1.4}$$

One way of deriving the formulas for $\rho_{i+1/2,j}^{n+1/2}$, $\rho_{i,j+1/2}^{n+1/2}$ is to notice that they are the averages of P^I over the region swept out by the characteristics through the zone edges centered, respectively, at $(i+\frac{1}{2}, j)$ and $(i, j+\frac{1}{2})$ (Fig. 2). We shall refer to this scheme as the corner transport upwind (CTU) scheme, since it takes into account the effect of information propagating across corners of zones in calculating the flux. This scheme is first-order accurate. It also satisfies a maximum principle, since $\rho_{i+1/2,j}^{n+1/2}$, $\rho_{i,j+1/2}^{n+1/2}$ are weighted sums, with nonnegative weights, of values of the solution at time t^n.

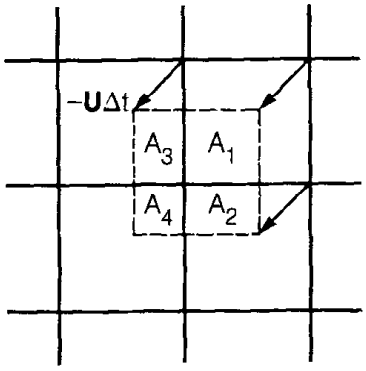


FIG. 1. The region over which we average ρ^I to obtain the new value for ρ is outlined with a dotted line. It is obtained by following the integral curves of the vector field $\mathbf{u}$ (in this case, straight lines) backwards in time by Δt from points in Δ_{ij}.

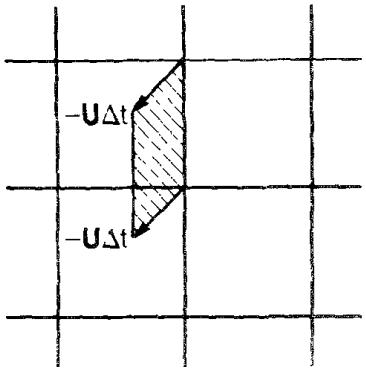


FIG. 2. The shaded region is the region over which one averages ρ^l to obtain the CTU flux at the zone edge bounding that region. It is the set of all points from which characteristics can reach that zone edge between time t^n and $t^n + \Delta t$.

One fact that is immediately seen from the formula given above for the floxes is the difference between the CTU scheme and the conventional donor cell differencing. In the latter case, $\rho^{n+1/2}_{i-1/2,j} = \rho^n_{i,j}$, $\rho^{n+1/2}_{i,j-1/2} = \rho^n_{i,j}$. Thus, in this scheme, we are adding a time-centered correction term to the donor-cell flux which estimates the effect on the flux of the gradients in the transverse direction. This corresponds to subtracting from the donor cell algorithm a term which, to leading order in the truncation error, is always destabilizing. This is reflected in the differing CFL time step limits for the two schemes:

$$\text{CTU: } \max\left(\frac{u\,\Delta t}{\Delta x}, \frac{v\,\Delta t}{\Delta y}\right) \leqslant 1. \tag{1.5}$$

$$\text{Donor-cell: } \quad \frac{u\,\Delta t}{\Delta x} + \frac{v\,\Delta t}{\Delta y} \leqslant 1, \tag{1.6}$$

where (1.5) is a sufficient condition, and (1.6) is a necessary condition, as is easily checked using Fourier analysis.

One can view schemes of the form (1.3)–(1.4) as being predictor-corrector schemes. One regards the calculation of $\rho^{n+1/2}_{i+1/2,j}$, $\rho^{n+1/2}_{i,j+1/2}$ as the predictor step, with the conservative differencing as the corrector step. Thus, if $\rho^{n+1/2}_{i+1/2,j}$ were to be calculated in such a way as to have a local truncation error of $O(\Delta t^2)$ in smooth regions, then the scheme would be second-order accurate. To obtain such an estimate for $\rho^{n+1/2}_{i+1/2,j}$ one must have

$$\begin{aligned}
\rho^{n+1/2}_{i+1/2,j} &= \rho^n_{i,j} + \frac{\Delta t}{2}\frac{\partial p}{\partial t} + \frac{\Delta x}{2}\frac{\partial \rho}{\partial x} \\
&= \rho^n_{i,j} - \frac{\Delta t}{2}\left(u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y}\right) + \frac{\Delta x}{2}\frac{\partial \rho}{\partial x} \\
&= \rho^n_{i,j} + \left(\frac{\Delta x}{2} - \frac{u\,\Delta t}{2}\right)\frac{\partial \rho}{\partial x} - \frac{v\,\Delta t}{2}\frac{\partial \rho}{\partial y}.
\end{aligned} \tag{1.7}$$

The only terms in (1.7) missing for the CTU flux (1.4) are the ones involving $\partial\rho/\partial x$. Thus, we add that term to $\rho^{n+1/2}_{i+1/2,j}$ to obtain a second-order flux:

$$\rho^{n+1/2}_{i+1/2,j} = \rho^n_{i,j} + \left(\frac{\Delta x}{2} - u\frac{\Delta t}{2}\right)\frac{\Delta^x \rho_{i,j}}{\Delta x} - \frac{v\,\Delta t}{2\,\Delta y}(\rho^n_{i,j} - \rho^n_{i,j-1}). \tag{1.8}$$

Here $\Delta^x \rho_{i,j}/\Delta x$ should be a difference approximation to $(\partial\rho/\partial x)|_{(i\Delta x, j\Delta y)}$, and $\Delta^x\rho$ should also be limited to suppress oscillations at discontinuities. The simplest choice is a central difference approximation to $(\partial\rho/\partial x)$, with the 1-dimensional limiter given in [20]:

$$\begin{aligned}(\Delta^x\rho)_{i,j} = \min(&\tfrac{1}{2}|\rho^n_{i+1,j} - \rho^n_{i-1,j}|, 2|\rho^n_{i+1,j} - \rho^n_{i,j}|, 2|\rho^n_{i,j} - \rho^n_{i-1,j}|)\\ &\times \operatorname{sgn}(\rho^n_{i+1,j} - \rho^n_{i-1,j}) \qquad \text{if} \quad (\rho^n_{i+1,j} - \rho^n_{i,j})(\rho^n_{i,j} - \rho^n_{i-1,j}) > 0;\\ &= 0 \qquad \text{otherwise.}\end{aligned} \tag{1.9}$$

Similarly, we define

$$\rho^{n+1/2}_{i,j+1/2} = \rho^n_{i,j} + \left(\frac{\Delta y}{2} - \frac{\Delta t}{2}v\right)\frac{(\Delta^y\rho)_{i,j}}{\Delta y} - \frac{\Delta t}{2\,\Delta x}u(\rho^n_{i,j} - \rho^n_{i-1,j}),$$

where $\Delta^y\rho$ is a monotonized central difference formula, such as the one given by (1.9) with the roles of i and j reversed. Because of the nonlinear switch in the definition of $\Delta^x\rho$, $\Delta^y\rho$, one cannot perform a formal error analysis on this algorithm. However, in smooth regions, one expects $\Delta^x\rho$, $\Delta^y\rho$ to be given by the central difference operators $(\Delta^x\rho)_{i,j} = \frac{1}{2}(\rho_{i+1,j} - \rho_{i-1,j})$, $(\Delta^y\rho)_{i,j} = \frac{1}{2}(\rho_{i,j+1} - \rho_{i,j-1})$. In this case, one can perform the linear error analysis and find that the scheme is second-order accurate. We have also calculated the amplification factor and evaluated it numerically; we have found that, as long as the time step satisfies (1.5), the second-order algorithm does not amplify any Fourier modes.

There is not a great deal one can say about the monotonicity properties of this algorithm, save that, when the slopes are fully limited, i.e., $\Delta^y\rho = \Delta^x\rho = 0$, it reduces to the first-order CTU scheme described above. In order to have this property, it is necessary to treat the spatial derivatives in the predictor step in a non-symmetric way: the derivatives in the direction tangent to the zone edge are approximated by upwind differences, and are not subject to monotonicity constraints, while the derivatives in the direction normal to the zone edge are approximated by monotonized central differences. For linear advection of a discontinuity oblique to the grid, the algorithm appears to produce monotone results.

A different approach to the one taken here, more in line with the geometric constructions in [20], would be to construct piecewise linear interpolants of ρ, suitably monotonized, and to integrate over surfaces swept out by the characteritics to obtain fluxes, similar to what was done to obtain the flux form (1.4) for the CTU scheme. We have not done so here: for a development along such lines, see [21]. However, for strongly nonlinear problems, we find that a somewhat more elaborate

treatment of the transverse derivatives than simply using first-order upwind differencing will be required, leading to an algorithm which is intermediate in complexity. This algorithm will be discussed in the next section.

2. Systems of Conservation Laws

In this section, we will consider algorithms for solving numerically the initial value problem

$$\begin{gathered}\frac{\partial U}{\partial t}+\nabla\cdot\mathbf{F}=0\\ U(\mathbf{x},t)=U:\mathbf{R}^2\times[0,T]\to\mathbf{R}^M\\ \mathbf{F}=(F^x,F^y)\in\mathbf{R}^M\times\mathbf{R}^M\\ U(\mathbf{x},0)=U_0(\mathbf{x}).\end{gathered}\tag{2.1}$$

For each $\mathbf{n}\in\mathbf{R}^2$ we define the projected equations (along $\mathbf{n}$) to be the 1-dimensional system of conservation laws

$$\frac{\partial U}{\partial t}+\frac{\partial F^{\mathbf{n}}}{\partial \chi}=0\qquad F^{\mathbf{n}}(U)=\mathbf{n}\cdot\mathbf{F}(U).\tag{2.2}$$

We say that the system (2.1) is hyperbolic if, for every $\mathbf{n}$ the projected equations

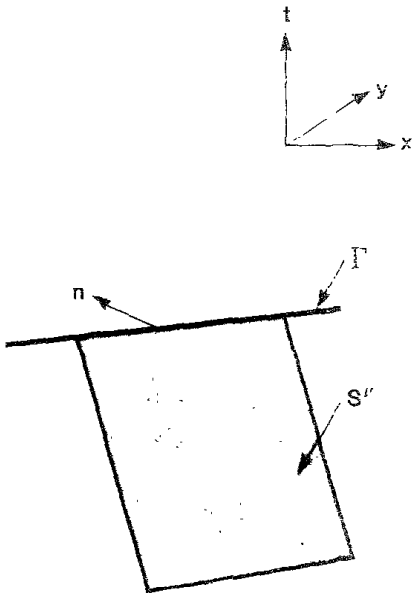


FIG. 3. Characteristic surfaces in two space dimensions. Γ is a curve in the spatial plane with normal vector field $\mathbf{n}$, and S^ν is one of the M characteristic surfaces in space-time passing through Γ.

(2.2) are hyperbolic, i.e., that the linearized coefficient matrix $\nabla_U F^{\mathbf{n}} = A^{\mathbf{n}}$ has M real eigenvalues $\lambda^{n,1} \leqslant \cdots \leqslant \lambda^{n,M}$ corresponding to M linearly independent left and right eigenvectors $(l^{\mathbf{n},\nu}, r^{\mathbf{n},\nu})$, $\nu = 1, \ldots, M$. We also have $A^{\mathbf{n}} = \mathbf{n} \cdot \mathbf{A}$, where $\mathbf{A} = (A^x, A^y)$, $A^x = \nabla_U F^x$, $A^y = \nabla_U F^y$. The left and right eigenvectors can be chosen so as to be biorthonormal, i.e., $l^{\mathbf{n},\nu} \cdot r^{\mathbf{n},\nu'} = \delta_{\nu,\nu'}$, so that the expansion of a vector $w \in \mathbf{R}^M$ in terms of the $r^{\mathbf{n},\nu}$'s is given by $w = \sum_{\nu=1,\ldots,M} \alpha^{\mathbf{n},\nu} r^{\mathbf{n},\nu}$, with $\alpha^{\mathbf{n},\nu} = l^{\mathbf{n},\nu} \cdot w$.

Our algorithm for the calculation of conservative fluxes is motivated in part by a version of the multidimensional theory of characteristics, which we review briefly here; for a more extensive discussion, see [8, 16]. If Γ is a curve in the plane $\{(\mathbf{x}, t): t = t_0\}$, then there exist surfaces $S^1, \ldots, S^M$ called characteristic surfaces, passing through Γ, such that the normal to S^ν at a point $(\mathbf{x}, t)$ is of the form $(\mathbf{n}, -\lambda^{\mathbf{n},\nu})$, where $\lambda^{\mathbf{n},\nu}$ is the νth eigenvalue of the projected equations in the direction of the unit vector $\mathbf{n}$ (see Fig. 3). The significance of these surfaces is that along each of these surfaces, a continuous, piecewise C^1 solution to (2.1) satisfies the following interior partial differential relation:

$$
\begin{aligned}
0 &= l^{\mathbf{n},\nu} \cdot \left(\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U \right) \\
&= l^{\mathbf{n},\nu} \cdot \left(\frac{\partial U}{\partial t} + (\mathbf{n} \cdot \mathbf{A})(\mathbf{n} \cdot \nabla U) + (\mathbf{t} \cdot \mathbf{A})(\mathbf{t} \cdot \nabla U) \right) \\
&= l^{\mathbf{n},\nu} \cdot \left(\frac{\partial U}{\partial t} + \lambda^{\mathbf{n},\nu} \mathbf{n} \cdot \nabla U + (\mathbf{t} \cdot \mathbf{A})(\mathbf{t} \cdot \nabla U) \right),
\end{aligned} \tag{2.3}
$$

where $\mathbf{t}$ is a unit vector orthogonal to $\mathbf{n}$ in the plane. Since $(\lambda^{\mathbf{n},\nu}\mathbf{n}, 1)$ and $(\mathbf{t}, 0)$ are tangent to S^ν, then (2.3) contains only derivatives in directions tangent to S^ν. In particular, if we define $d/d\sigma^\nu$ to be differentiation in the direction of the vector field $(\lambda^{\mathbf{n},\nu}\mathbf{n}, 1)$, then (2.3) becomes

$$
l^{\mathbf{n},\nu} \cdot \frac{dU}{d\sigma^\nu} + (l^{\mathbf{n},\nu} \cdot A^{\mathbf{t}})(\mathbf{t} \cdot \nabla U) = 0; \tag{2.4}
$$

i.e., we obtain the ordinary differential relation from the theory of characteristics in one dimension for the system projected in the $\mathbf{n}$ direction, with the derivatives in the $\mathbf{t}$ direction acting as source terms.

Finally, we assume that the Riemann problem for the projected system (2.2) is well posed for all $\mathbf{n} \in \mathbf{R}^2$, i.e., that the initial value problem for (2.2) given by

$$
\begin{aligned}
U(\chi, 0) &= U_L \qquad \text{for} \quad \chi < 0 \\
&= U_R \qquad \text{for} \quad \chi > 0
\end{aligned}
$$

has a unique solution with appropriate entropy conditions, for any choice of U_L, U_R for which (2.2) is hyperbolic. This solution is a function only of the similarity

variable χ/t; throughout this paper, when we require the solution to a Riemann problem, it will be at the point $\chi/t = 0$.

We assume, as in the scalar case, that we know $U^n_{i,j}$, the average of the solution over $\Delta_{i,j}$, the zone centered at $(i\,\Delta x, j\,\Delta y)$:

$$U^n_{i,j} = \frac{1}{\sigma_{i,j}} \int_{\Delta_{i,j}} U(\mathbf{x}, t^n)\, d\mathbf{x}.$$

We want to extend the algorithm described in the previous section to calculate $U^{n+1}_{i,j}$. The difficulty here is that the different modes of wave propagation can carry gradient information from different sides of the zone edge where the flux is to be evaluated. We solve this problem by using predictor calculations similar to (1.8) to calculate two states at a zone edge, representing the propagation of signals coming from the left and the right of the zone edge. We then obtain a single value for the flux by solving a Riemann problem given the two states, with the jump assumed to be parallel to the zone edge.

The algorithm can be broken up into the following four steps:

(1) the calculation of monotonized central difference approximations to

$$\frac{\Delta^x U}{\Delta x} \approx \left.\frac{\partial U}{\partial x}\right|_{(i\Delta x, j\Delta y)}, \qquad \frac{\Delta^y U}{\Delta y} \approx \left.\frac{\partial U}{\partial y}\right|_{(i\Delta x, j\Delta y)};$$

(2) the construction of time-centered left and right states at the zone edges: $U^{n+1/2}_{i+1/2,j,L}$, $U^{n+1/2}_{i+1/2,j,R}$ at $((i+\frac{1}{2})\,\Delta x, j\,\Delta y)$, and $U^{n+1/2}_{i,j+1/2,L}$, $U^{n+1/2}_{i,j+1/2,R}$ at $(i\,\Delta x, (j+\frac{1}{2})\,\Delta y)$;

(3) the solution of the Riemann problem at the zone edges for the projected equations along the normal to that zone edge, given the left and right states computed in (2), to obtain $U^{n+1/2}_{i+1/2,j}$, $U^{n+1/2}_{i,j+1/2}$;

(4) the conservative differencing of the fluxes $F^x_{i+1/2,j} = F^x(U^{n+1/2}_{i+1/2,j})$, $F^y_{i,j+1/2} = F^y(U^{n+1/2}_{i,j+1/2})$ to obtain $U^{n+1}_{i,j}$:

$$U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x}(F^x_{i-1/2,j} - F^x_{i+1/2,j}) + \frac{\Delta t}{\Delta y}(F^y_{i,j-1/2} - F^y_{i,j+1/2}).$$

We will describe the details of only the calculation of $F^x_{i+1/2,j}$; the other fluxes are calculated along the same lines, interchanging the roles of i and j, x and y.

The calculation of slopes follows the pattern seen in the scalar case: we use central difference to approximate the spatial derivatives of U and constrain them using a 1-dimensional monotonicity algorithm. In imposing monotonicity constraints, there are two strategies which have been used successfully in one dimension. The first is to perform a nonlinear change of variables such that the new dependent variables are the Riemann invariants, i.e., a set of variables $(v^1, \ldots, v^M)^T$ such that $l^\nu \cdot \nabla_U v^\mu = \delta_{\nu\mu}$, and interpolate those variables componentwise using monotonized

interpolation such as the one given for the scalar case in the previous section. This procedure can be done only for special systems, since such a set of Riemann invariants does not, in general, exist when $M > 2$. A variation on this procedure is done for Euler's equations for compressible flow, where the primitive variables are interpolated; this is discussed in Section 4. The second approach, due to Harten [14], is to expand the central difference approximation to the spatial derivatives in terms of the right eigenvectors of the coefficient matrix of the linearized equation and constrain the amplitudes in that expansion. Since the latter procedure is well defined for general systems of conservation laws, we will describe it here.

To calculate $(\Delta^x U)_{i,j}$ we define the expansions,

$$\begin{aligned} \tfrac{1}{2}(U_{i+1,j} - U_{i-1,j}) &= \sum \alpha_C^\nu r^{x,\nu}, \\ 2(U_{i+1,j} - U_{i,j}) &= \sum \alpha_R^\nu r^{x,\nu}, \\ 2(U_{i,j} - U_{i-1,j}) &= \sum \alpha_L^\nu r^{x,\nu}, \end{aligned} \tag{2.5}$$

where $l^{x,\nu}$, $r^{x,\nu}$, $\lambda^{x,\nu}$ are the eigenvectors and eigenvalues of the equations projected in the x coordinate direction. Then $(\Delta^x U)_{i,j}$ is given by

$$(\Delta^x U)_{i,j} = \sum \alpha^\nu r^{x,\nu} \tag{2.6}$$

$$\begin{aligned} \alpha^\nu &= \min(|\alpha_C^\nu|, |\alpha_L^\nu|, |\alpha_R^\nu|) \times \operatorname{sgn}(\alpha_C^\nu) \qquad \text{if} \quad \alpha_L^\nu \alpha_R^\nu > 0 \\ &= 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \text{otherwise.} \end{aligned}$$

Next, we define the left and right states at the zone edges $U^{n+1/2}_{i+1/2,j,L}$, $U^{n+1/2}_{i+1/2,j,R}$. We extrapolate from the zone centers on either side of the zone edge at $((i+\frac{1}{2})\Delta x, j\Delta y)$, using a formula similar to (1.7):

$$\begin{aligned} U^{n+1/2}_{i+1/2,j,S} &= U^n_{i+k,j} \pm \frac{\Delta x}{2}\frac{\partial U}{\partial x} + \frac{\Delta t}{2}\frac{\partial U}{\partial t} \\ &= U^n_{i+k,j} \pm \frac{\Delta x}{2}\frac{\partial U}{\partial x} - \frac{\Delta t}{2}\left(\frac{\partial F^x}{\partial x} + \frac{\partial F^y}{\partial y}\right) \\ &= U^n_{i+k,j} + \left(\pm\frac{\Delta x}{2} - \frac{\Delta t A^x}{2}\right)\frac{\partial U}{\partial x} - \frac{\Delta t}{2}\frac{\partial F^y}{\partial y}. \end{aligned} \tag{2.7}$$

Here, and in what follows, we use expressions such as (2.7) involving the symbols $(S, \pm, k)$ to mean a pair of expressions: one with $(S, \pm, k)$ replaced by $(L, +, 0)$, the other with $(S, \pm, k)$ replaced by $(R, -, 1)$. In calculating $U^{n+1/2}_{i+1/2,j,S}$, we approximate $\partial U/\partial x$ by the monotonized central differences $\Delta^x U/\Delta x$ and the $\partial F^y/\partial y$ term by a difference of Godunov fluxes, the extension to nonlinear systems in one dimension of upwind differencing for linear scalar equations.

It is convenient to view the calculation of $U^{n+1/2}_{i+1/2,j,L}$, $U^{n+1/2}_{i+1/2,j,R}$ as consisting of two steps, the first involving the monotonized central difference approximations to $\partial U/\partial x$, the second involving the transverse derivatives:

$$\hat{U}_{i+1/2,j,S} = U^n_{i+k,j} + \left(\pm\frac{\Delta x}{2} - \frac{\Delta t}{2}A^x\right)\frac{\partial U}{\partial x} \tag{2.8}$$

$$U^{n+1/2}_{i+1/2,j,S} = \hat{U}_{i+1/2,j,S} - \frac{\Delta t}{2}\frac{\partial F^y}{\partial y}. \tag{2.9}$$

In order to calculate $\hat{U}_{i+1/2,j,S}$ for linear problems, it would suffice simply to replace $\partial U/\partial x$ by $(\Delta^x U)_{i,j}/\Delta x$. However, we make two changes in (2.8), which, for constant coefficient problems, are redundant operations leading to identical values for $U^{n+1/2}_{i+1/2,j}$, but which have been seen to lead to a somewhat more robust algorithm for strongly nonlinear problems. This first is to discard in the $\partial U/\partial x$ term the components corresponding to characteristics which do not propagate towards the zone edge. The second is to introduce arbitrary reference states $\tilde{U}_L$, $\tilde{U}_R$, taking advantage of the fact that the characteristic projection operators appearing in both the construction of the left and right states and in the solution of the Riemann problem act on increments of U. The resulting algorithm is given as follows:

$$\hat{U}_{i+1/2,j,S} = \tilde{U}_S + P_S(U^n_{i+k,j} - \tilde{U}_S) + P_S\left(\pm\frac{1}{2} - \frac{\Delta t}{2\Delta x}A^x(U_{i+k,j})\right)(\Delta^x U)_{i+k,j} \tag{2.10}$$

$$P_S w = \sum_{w^v \pm \lambda^{x,v}(U_{i+k,j})>0} (l^{x,v}_{i+k,j}\cdot w)\, r^{x,v}_{i+k,j}.$$

The reference states $\tilde{U}_L$, $\tilde{U}_R$ are chosen so as to reduce to as great an extent as possible the size of the sum of the terms multiplied by the characteristic projection operators P_S. One possibility is to take

$$\begin{aligned}\tilde{U}_L &= U^n_{i,j} + \left(\frac{1}{2} - \max(\lambda^{x,M}(U_{i,j}), 0)\frac{\Delta t}{2\Delta x}\right)\Delta^x U_{i,j} \\ \tilde{U}_R &= U^n_{i+1,j} - \left(\frac{1}{2} + \min(\lambda^{x,1}(U_{i+1,j}), 0)\frac{\Delta t}{2\Delta x}\right)\Delta^x U_{i+1,j}.\end{aligned} \tag{2.11}$$

The additional cost of applying the characteristic projection operators is small. Because of the monotonicity algorithm, we already know the expansion of $\Delta^x U$ in terms of the right eigenvectors. Applying the characteristic projection operators to $(\Delta^x U)$ is accomplished by setting to zero the coefficients of the eigenvector expansion of $(\Delta^x U)$ which have associated propagation speeds with the wrong sign. Finally, the calculation of the terms involving A^x is easily accomplished using the fact that the projection operators are sums of eigenprojections of A^x, implying that $P_S A^x \Delta^x U = \sum_{\pm\lambda^{x,v}>0} \lambda^{x,v}\alpha^v r^{x,v}$. Using this fact, and with the above choice of $\tilde{U}_L$, $\tilde{U}_R$, we obtain the following explicit expression for (2.10):

$$\hat{U}_{i+1/2,j,L} = \tilde{U}_L + \frac{\Delta t}{2\Delta x} \sum_{\nu:\lambda^{x,\nu}_{i,j}>0} (\lambda^{x,M}_{i,j} - \lambda^{x,\nu}_{i,j})\, \alpha^{x,\nu}_{i,j} r^{x,\nu}_{i,j}$$
$$\hat{U}_{i+1/2,j,R} = \tilde{U}_R + \frac{\Delta t}{2\Delta x} \sum_{\nu:\lambda^{x,\nu}_{i+1,j}<0} (\lambda^{x,1}_{i+1,j} - \lambda^{x,\nu}_{i+1,j})\, \alpha^{x,\nu}_{i+1,j} r^{x,\nu}_{i+1,j}, \tag{2.12}$$

where the $\alpha^{x,\nu}_{i,j}$'s are the expansion coefficients of $(\Delta^x U)_{i,j}$ given by (2.6). This procedure is essentially that given in [7] for computing the left and right states for the 1-dimensional algorithm, applied to the case of piecewise linear interpolation.

To complete the calculation of $U^{n+1/2}_{i+1/2,j,S}$ we approximate $(\partial F^y/\partial y)|_{(i\Delta x, j\Delta y)}$ by some appropriate upwind flux difference. The simplest choice is to use Godunov's first-order method to evaluate $\partial F^y/\partial y$. If we define $U^T_{i,j+1/2}$ to be the solution to the Riemann problem for the projected equations along the y-direction, with left and right states

$$(U^{\mathrm{T}}_{i,j+1/2,L},\, U^{\mathrm{T}}_{i,j+1/2,R}) = (U^n_{i,j},\, U^n_{i,j+1}) \tag{2.13}$$

then

$$U^{n+1/2}_{i+1/2,j,S} = \hat{U}_{i+1/2,j,S} - \frac{\Delta t}{2\Delta y}(F^y(U^{\mathrm{T}}_{i+k,j+1/2}) - F^y(U^{\mathrm{T}}_{i+k,j-1/2})) \tag{2.14}$$

is a sufficiently accurate approximation to (2.9) to yield an algorithm that is second-order accurate. For problems involving moderately strong nonlinear discontinuities which are oblique to the mesh directions, it is necessary to use a slightly more complicated algorithm to evaluate the effect of the transverse derivative term $(\partial F^y/\partial y)(\Delta t/2)$ on the left and right states. This term estimates the change in the solution due to the y-gradients. In the case of an oblique discontinuity, if the estimate is sufficiently different from the actual change calculated in the conservation step, the solution will overshoot, or the discontinuity will spread, depending on the relative signs of the gradient and the error. To alleviate this problem, we use an estimate for $\partial F^y/\partial y$ which is closer to what we will actually use in the conservation step, by taking $U^{\mathrm{T}}_{i,j+1/2}$ to be the solution to the Riemann problem for the equations projected along the y-direction with left and right states

$$(U^{\mathrm{T}}_{i,j+1/2,L},\, U^{\mathrm{T}}_{i,j+1/2,R}) = (\hat{U}_{i,j+1/2,L},\, \hat{U}_{i,j+1/2,R}), \tag{2.15}$$

where $\hat{U}_{i,j+1/2,L}$, $\hat{U}_{i,j+1/2,R}$ is computed using the analogue of (2.10) for the zone edge at $(i\,\Delta x, (j+\frac{1}{2})\,\Delta y)$.

Given the left and right states defined as above, we solve the Riemann problem for the 1-dimensional equation projected along the x direction to obtain $U^{n+1/2}_{i+1/2,j}$. In the case of constant coefficient equations, it is easy to check that $U^{n+1/2}_{i+1/2,j}$ satisfies the following linear equations, independent of the choice of $\tilde{U}_L$, $\tilde{U}_R$:

$$l^{x,\nu} \cdot (U^{n+1/2}_{i+1/2,j} - U_{i+1/2,j,\nu}) - \frac{\Delta t}{2\Delta y} l^{x,\nu} \cdot (F^y(U^{\mathrm{T}}_{i+k,j+1/2}) - F^y(U^{\mathrm{T}}_{i+k,j-1/2})) = 0, \tag{2.16}$$

where

$$U_{i+1/2,j,v} = U^n_{i,j} + \left(\frac{1}{2} - \lambda^{x,v}\frac{\Delta t}{2\,\Delta x}\right)(\Delta^x U)_{i,j}, \qquad \text{if } \lambda^{x,v} > 0$$

$$= U^n_{i+1,j} - \left(\frac{1}{2} + \lambda^{x,v}\frac{\Delta t}{2\,\Delta x}\right)(\Delta^x U)_{i+1,j}, \qquad \text{otherwise.}$$

This is a finite difference approximation to the characteristic form of Eqs. (2.4) on the M characteristic surfaces intersecting the line $\{(x, y): x = (i+\frac{1}{2})\,\Delta x\}$ at time $t^{n+1/2}$. The proof is a routine calculation using the characteristic projection operators; the key fact that is required is that the solution to the Riemann problem for (2.2) with left and right states W_L, W_R is given by

$$W = P_L W_L + P_R W_R,$$

where P_L, P_R are the projection operators defined in (2.10). In the case where the equations are nonlinear, but the solutions are smooth, $U^{n+1/2}_{i+1/2,j}$ satisfies (2.16) modulo terms which are second order in the mesh spacing, provided that $\tilde{U}_S - U^n_{i+k,j}$ is of the order of the mesh spacing, where the eigenvectors and eigenvalues are evaluated at $U^{n+1/2}_{i+1/2,j}$. This fact describes one sense in which the algorithm described here is upstream-centered for smooth solutions: the value of the predictor $U^{n+1/2}_{i+1/2,j}$ is given as a solution to M linear equations which are finite difference approximations to the characteristic equations.

Finally, we need to specify a bound on the time step for stability. We expect that the CFL condition should be given by

$$\max_{i,j,v}\left(\left|\lambda^{x,v}_{i,j}\frac{\Delta t}{\Delta x}\right|, \left|\lambda^{y,v}_{i,j}\frac{\Delta t}{\Delta y}\right|\right) \leqslant 1, \tag{2.17}$$

by analogy with the stability condition (1.5) for the advection equation. In the case where A^x and A^y commute, the above stability condition holds in the sense that it held for the scalar equation, i.e., that the fully limited scheme, and the scheme without limiting, both have (2.17) as necessary and sufficient conditions for Fourier stability. This follows easily from the analogous result for scalar equations, plus the fact that the system can be diagonalized. We have not proven (2.17) for any problem for which A^x and A^y do not commute. However, we have used the above condition as a time step control for our gas dynamics calculations and have seen no evidence of instability.

3. Quadrilateral Grids

The above algorithm can be extended to the case of arbitrary quadrilateral grids. For the purposes of deriving the algorithm we will assume that our grid comes from

a smooth coordinate mapping, although the final difference algorithm will be expressed only in terms of differences between coordinates of the corners of the quadrilateral mesh.

We now assume that our computational domain is divided into quadrilaterals $\Delta_{i,j}$ with corners located at $(x_{i+1/2,j+1/2}, y_{i+1/2,j+1/2})$. Furthermore, we assume there is a smooth map $(\xi, \eta) \leftrightarrow (x, y)$ between some coordinate space and physical space, with a rectangular mesh in (ξ, η) space with corners located at $(\xi_{i+1/2}, \eta_{j+1/2})$ such that $(x_{i+1/2,j+1/2}, y_{i+1/2,j+1/2}) = (x(\xi_{i+1/2}, \eta_{j+1/2}), y(\xi_{i+1/2}, \eta_{j+1/2}))$. We can transform the system (2.1) to the (ξ, η) coordinate system:

$$\frac{\partial (JU)}{\partial t} + \frac{\partial F^{\xi}}{\partial \xi} + \frac{\partial F^{\eta}}{\partial \eta} = 0 \tag{3.1}$$

$$J = \mathrm{Det}(\nabla_{(\xi,\eta)}(x, y))$$

$$F^{\xi} = \mathbf{n}^{\eta} \cdot \mathbf{F}, \qquad F^{\eta} = \mathbf{n}^{\xi} \cdot \mathbf{F}$$

$$\mathbf{n}^{\eta} = \left(\frac{\partial y}{\partial \eta}, -\frac{\partial x}{\partial \eta} \right), \qquad \mathbf{n}^{\xi} = \left(-\frac{\partial y}{\partial \xi}, \frac{\partial x}{\partial \xi} \right).$$

Without loss of generality we assume here that $J > 0$. We define finite difference approximations to the derivatives of the grid mapping function:

$$(\Delta^{\xi}\mathbf{x})_{i,j+1,2} = \mathbf{x}_{i+1/2,j+1/2} - \mathbf{x}_{i-1/2,j+1/2} \approx \left. \frac{\partial \mathbf{x}}{\partial \xi} \right|_{\xi_i, \eta_{j+1/2}} \Delta\xi_i$$

$$(\Delta^{\eta}\mathbf{x})_{i+1/2,j} = \mathbf{x}_{i+1/2,j+1/2} - \mathbf{x}_{i+1/2,j-1/2} \approx \left. \frac{\partial \mathbf{x}}{\partial \eta} \right|_{\xi_{i+1/2}, \eta_j} \Delta\eta_j$$

$$(\Delta^{\xi}\mathbf{x})_{i,j} = \tfrac{1}{2}((\Delta^{\xi}\mathbf{x})_{i,j+1/2} + (\Delta^{\xi}\mathbf{x})_{i,j-1/2}) \tag{3.2}$$

$$(\Delta^{\eta}\mathbf{x})_{i,j} = \tfrac{1}{2}((\Delta^{\eta}\mathbf{x})_{i+1/2,j} + (\Delta^{\eta}\mathbf{x})_{i-1/2,j})$$

$$\sigma_{i,j} = \tfrac{1}{2}((x_{i+1/2,j-1/2} - x_{i-1/2,j+1/2})(y_{i+1/2,j+1/2} - y_{i-1/2,j-1/2}) + (x_{i+1/2,j+1/2} - x_{i-1/2,j-1/2})(y_{i-1/2,j+1/2} - y_{i+1/2,j-1/2})).$$

Using these finite differences, we can make the connection between the mapping derivatives appearing in the transformed equations (3.1) and the geometry of the finite difference grid in physical space (Fig. 4): $\sigma_{ij} \approx J(\xi_i, \eta_j)\,\Delta\xi_i\,\Delta\eta_j$ is the area of the (i, j)th zone, and $\mathbf{n}^{\xi}\,\Delta\xi_i \approx -(\Delta^{\xi}\mathbf{x})^{\perp}_{i,j+1/2}$, $\mathbf{n}^{\eta}\,\Delta\eta_j \approx (\Delta^{\eta}\mathbf{x})^{\perp}_{i+1/2,j}$ are normal to the zone edges, where we use the notation $(w_1, w_2)^{\perp} = (w_2, -w_1)$.

As in the previous section, we will assume that, at time step n, we know $U^n_{i,j}$, the average of U over $\Delta_{i,j}$. The procedure for calculating $U^{n+1}_{i,j}$ follows the same basic outline as that for the rectangular grid case. We construct time-centered left and right states at the zone edges, solve the Riemann problem, and difference the fluxes conservatively, taking care that, at each step, the effect of the quadrilateral mesh is accounted for in a suitable fashion.

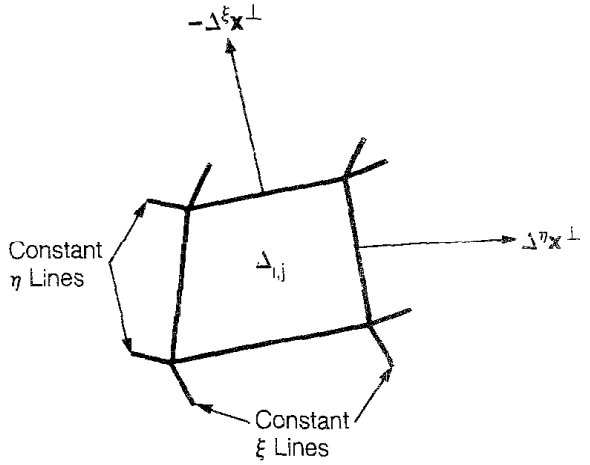


FIG. 4. Geometric interpretation of the difference approximations to the derivatives of the grid mapping.

Our conservative difference step will be of the "finite volume" type:

$$U_{i,j}^{n+1} = U_{i,j}^{n} + \frac{\Delta t}{\sigma_{i,j}}\left((\Delta^{\eta}\mathbf{x})^{\perp}_{i-1/2,j}\cdot\mathbf{F}(U^{n+1/2}_{i-1/2,j}) - (\Delta^{\eta}\mathbf{x})^{\perp}_{i+1/2,j}\cdot\mathbf{F}(U^{n+1/2}_{i+1/2,j})\right.$$
$$\left. - (\Delta^{\xi}\mathbf{x})^{\perp}_{i,j-1/2}\cdot\mathbf{F}(U^{n+1/2}_{i,j-1/2}) + (\Delta^{\xi}\mathbf{x})^{\perp}_{i,j+1/2}\cdot\mathbf{F}(U^{n+1/2}_{i,j+1/2})\right). \tag{3.3}$$

It is clear that this formula is a conservative finite difference approximation to (3.1). This formula can also be obtained by integrating (2.1) over $\Delta_{i,j}\times[t^n, t^{n+1}]$, applying the divergence theorem, and approximating the resulting surface integrals using the midpoint formula. From that point of view, each of the terms multiplied by $\Delta t/\sigma_{i,j}$ represents a time- and space-averaged flux through one of the edges of $\Delta_{i,j}$.

Our strategy for obtaining values for $U^{n+1/2}_{i+1/2,j}$, $U^{n+1/2}_{i,j+1/2}$ follows the pattern used in the rectangular grid case. We extrapolate time-centered left and right limiting states at the zone edges using (3.1). We then solve the Riemann problem using these states for Eqs. (2.1) projected in the direction of the normal to the zone edges in physical space. We consider, for example, the zone edge centered at $(i+1/2, j)$ and we wish to construct $U^{n+1/2}_{i+1/2,j,L}$, $U^{n+1/2}_{i+1/2,j,R}$, the left and right states at that zone edge. The starting point for this is to consider the extrapolation formulae analogous to (2.7) for the system (3.1):

$$U^{n+1/2}_{i+1/2,j,S} = U^{n}_{i+k,j} \pm \frac{\Delta\xi_{i+k}}{2}\frac{\partial U}{\partial\xi} + \frac{\Delta t}{2}\frac{\partial U}{\partial t}$$
$$= U^{n}_{i+k,j} \pm \frac{\Delta\xi_{i+k}}{2}\frac{\partial U}{\partial\xi} - \frac{\Delta t}{2J}\left(\frac{\partial F^{\xi}}{\partial\xi} + \frac{\partial F^{\eta}}{\partial\eta}\right)$$
$$= U^{n}_{i+k,j} + \left(\pm\frac{1}{2} - \frac{\Delta t}{2J\Delta\xi_{i+k}}A^{\xi}\right)\frac{\partial U}{\partial\xi}\Delta\xi_{i+k} - \frac{\Delta t}{2J}\frac{\partial\mathbf{n}^{\eta}}{\partial\xi}\cdot\mathbf{F} - \frac{\Delta t}{2J}\frac{\partial F^{\eta}}{\partial\eta}, \tag{3.4}$$

where $A^\xi = \mathbf{n}^\eta \cdot \mathbf{A}$. The term $(\Delta t/2J)(\partial \mathbf{n}^\eta/\partial \xi)\cdot \mathbf{F}$ comes from putting $\partial F^\xi/\partial \xi$ in non-conservation form and is equal to zero in the rectangular grid case. We break this procedure into two steps:

$$\hat{U}_{i+1/2,j,S} = U^n_{i+k,j} + \left(\pm\frac{1}{2} - \frac{\Delta t}{2J\,\Delta\xi_{i+k}} A^\xi\right)\frac{\partial U}{\partial \xi}\Delta\xi_{i+k} \tag{3.5}$$

$$U^{n+1/2}_{i+1/2,j,S} = \hat{U}_{i+1/2,j,S} - \frac{\Delta t}{2J}\left(\frac{\partial \mathbf{n}^\eta}{\partial \xi}\cdot\mathbf{F} + \frac{\partial F^\eta}{\partial \eta}\right). \tag{3.6}$$

We approximate $\partial U/\partial\xi$ by monotonized central differences and $\partial F^\eta/\partial\eta$ by upwind differences. The term $(\partial \mathbf{n}^\eta/\partial\xi)\cdot\mathbf{F}$ is differenced in such a way as to exactly cancel the difference approximation to $\partial F^\eta/\partial\eta$ if there are no gradients in the η direction.

We first consider the calculation of $\hat{U}_{i+1/2,j,S}$. We approximate

$$\left(\pm\frac{1}{2} - \frac{\Delta t}{2J\,\Delta\xi_{i+k}} A^\xi\right) \approx \left(\pm\frac{1}{2} - \frac{\Delta t}{2\sigma_{i+k,j}} (\Delta^\eta \mathbf{x})^\perp_{i+k,j}\cdot \mathbf{A}(U^n_{i+k,j})\right), \tag{3.7}$$

where we have replaced J and $\mathbf{n}^\xi$, $\mathbf{n}^\eta$ by the appropriate difference approximations from (3.2). By analogy with the rectangular grid case, we want to approximate $(\partial U/\partial\xi)\,\Delta\xi_i$ with $(\Delta^\xi U)_{i,j}$, a central difference approximation to which some form of monotonicity constraint has been applied. If the coordinate mapping is smooth, then the formula (2.5) for equally spaced zones can be used without modification, while retaining second-order accuracy in regions where the solution is smooth. However, we replace the eigenvectors in the monotonicity constraints in (2.6) by $(l^{\xi,v}_{i,j}, r^{\xi,v}_{i,j})$, $v = 1, \ldots, M$, the left and right eigenvectors corresponding to the eigenvalues $\lambda^{\xi,1}_{i,j} \leqslant \cdots \leqslant \lambda^{\xi,M}_{i,j}$ of $(\Delta^\eta \mathbf{x})^\perp_{i,j}\cdot\mathbf{A}(U^n_{i,j})$. As before, we can also discard terms in (3.7) corresponding to signals propagating away from the zone edge and allow for an arbitrary choice of reference state $\tilde{U}_S$, obtaining the following analogue of (2.10) for a general quadrilateral grid:

$$\begin{aligned}\hat{U}_{i+1/2,j,S} = {} & \tilde{U}_S + P_S(U^n_{i,j} - \tilde{U}_S) \\ & + P_S\left(\pm\frac{1}{2} - \frac{\Delta t}{2\sigma_{i+k,j}} (\Delta^\eta \mathbf{x})^\perp_{i+k,j}\cdot\mathbf{A}(U^n_{i+k,j})\right)\cdot(\Delta^\xi U)_{i+k,j},\end{aligned} \tag{3.8}$$

where

$$P_S w = \sum_{v:\, \pm\lambda^{\xi,v}_{i+k,j} > 0} (l^{\xi,v}_{i+k,j}\cdot w)\, r^{\xi,v}_{i+k,j}.$$

We approximate $(\Delta t/2J)(\partial F^\eta/\partial\eta)$ by an appropriate upwind difference approximation. In general, it is of the form of the corresponding difference approximation in the conservative difference step (3.3):

$$-\frac{\Delta t}{2J}\frac{\partial F^\eta}{\partial \eta} \approx \frac{\Delta t}{2\sigma_{i,j}}\left((\Delta^\xi \mathbf{x})^\perp_{i,j+1/2}\cdot\mathbf{F}(U^{\mathrm{T}}_{i,j+1/2}) - (\Delta^\xi \mathbf{x})^\perp_{i,j-1/2}\cdot\mathbf{F}(U^{\mathrm{T}}_{i,j-1/2})\right). \tag{3.9}$$

Here $U^{\mathrm{T}}_{i,j+1/2}$ is calculated by solving a Riemann problem for the projected equations along $-(\Delta^{\xi}\mathbf{x})^{\perp}_{i,j+1/2}$ with left and right states $(U^{\mathrm{T}}_{i,j+1/2,L}, U^{\mathrm{T}}_{i,j+1/2,R})$. As in the rectangular grid case, $U^{\mathrm{T}}_{i,j+1/2,S}$ may be set to $U^{n}_{i,j+1}$ or $\hat{U}_{i,j+1/2,S}$. Finally, we approximate $(\Delta t/2J)(\partial \mathbf{n}^{\eta}/\partial\eta)\cdot\mathbf{F}$ using the finite difference approximations (3.2):

$$\frac{\Delta t}{2J}\frac{\partial \mathbf{n}^{\eta}}{\partial \xi}\cdot\mathbf{F} \approx \frac{\Delta t}{2\sigma_{i,j}}\left((\Delta^{\eta}\mathbf{x})^{\perp}_{i+1/2,j}-(\Delta^{\eta}\mathbf{x})^{\perp}_{i-1/2,j}\right)\cdot\mathbf{F}(U^{n}_{i,j}). \tag{3.10}$$

Collecting our difference approximations, our final value for $U^{n+1/2}_{i+1/2,j,S}$ is given by

$$\begin{aligned} U^{n+1/2}_{i+1/2,j,S} = \hat{U}_{i+1/2,j,S} + \frac{\Delta t}{2\sigma_{i,j}}\Big[&(\Delta^{\xi}\mathbf{x})^{\perp}_{i+k,j+1/2}\cdot\mathbf{F}(U^{\mathrm{T}}_{i+k,j+1/2}) \\ &-(\Delta^{\xi}\mathbf{x})^{\perp}_{i+k,j-1/2}\cdot\mathbf{F}(U^{\mathrm{T}}_{i+k,j-1/2}) \\ &-\left((\Delta^{\eta}\mathbf{x})^{\perp}_{i+1/2+k,j}-(\Delta^{\eta}\mathbf{x})^{\perp}_{i-1/2+k,j}\right)\cdot\mathbf{F}(U^{n}_{i+k,j})\Big]_{k}. \end{aligned} \tag{3.11}$$

We obtain $U^{n+1/2}_{i+1/2,j}$ by solving the Riemann problem for the projected equations along $(\Delta^{\eta}\mathbf{x})^{\perp}_{i+1/2,j}$ with left and right states $U^{n+1/2}_{i+1/2,j,L}$, $U^{n+1/2}_{i+1/2,j,R}$. $U^{n+1/2}_{i+1/2,j}$ satisfies finite difference approximations to the characteristic equations (2.4) for the characteristic surfaces through the $(i+1/2, j)$th zone edge in physical space, similar to (2.14).

The appropriate generalization of (2.17) as a CFL condition on the time step is given by

$$\max_{i,j,\nu}\left(\left|\lambda^{\xi,\nu}_{i,j}\frac{\Delta t}{\sigma_{i,j}}\right|, \left|\lambda^{\eta,\nu}_{i,j}\frac{\Delta t}{\sigma_{i,j}}\right|\right) \leqslant 1. \tag{3.12}$$

This is dimensionally correct since $\lambda^{\xi,\nu}_{i,j}$, $\lambda^{\eta,\nu}_{i,j}$ contain factors of $\Delta^{\eta}\mathbf{x}$, $\Delta^{\xi}\mathbf{x}$. In the case of advection, and if the coordinate transformation is a linear map, one can demonstrate by numerical evaluation of the Fourier transform, as was done for the rectangular mesh case, that this is the correct CFL condition. In general, the time step bound (3.12) has the following interpretation in terms of characteristics: Δt must be less than the time it takes a wave propagating in a direction normal to a zone edge to reach an opposite zone edge.

4. Gas Dynamics

We give in this section a detailed description of an algorithm of the type described above for the case of Euler's equations for inviscid compressible flow in two space variables, in planar geometry, on a general quadrilateral grid. The system we wish to solve is of the form (2.1), with $M=4$, and

$$U=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix},\qquad F^x(U)=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uE+up\end{pmatrix},\qquad F^y(U)=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vE+p\end{pmatrix}, \tag{4.1}$$

where ρ is the density, $(u, v)=\mathbf{u}$ the x and y components of velocity, and E the total energy per unit mass. The pressure is derived from these quantities via an equation of state, $p=p(\rho, e)$, where e is the internal energy per unit mass, given by $e=E-\frac{1}{2}(u^2+v^2)$. In this section, we will describe an algorithm suitable for use with a polytropic equation of state, i.e., for p given by $p(\rho, e)=\rho e(\gamma-1)$, and the adiabatic speed of sound c given by $c^2=\gamma p/\rho$. The case of a general convex equation of state is a straightforward extension of ideas in [6].

The projected equations for the system (4.1), are essentially those of gas dynamics in one dimension. If we project the equations in the $\mathbf{n}$ direction for $\mathbf{n}$ a unit vector, we can make a change of variables to obtain the following system equivalent to (2.2):

$$\frac{\partial W}{\partial t}+\frac{\partial G(W)}{\partial \chi}=0 \tag{4.2}$$

$$W=\begin{pmatrix}\rho\\ \rho u^{\mathrm{N}}\\ \rho u^{\mathrm{T}}\\ \rho E\end{pmatrix},\qquad G(W)=\begin{pmatrix}\rho u^{\mathrm{N}}\\ \rho(u^{\mathrm{N}})^2+p\\ \rho u^{\mathrm{N}}u^{\mathrm{T}}\\ \rho u^{\mathrm{N}}E+u^{\mathrm{N}}p\end{pmatrix}$$

Here $u^{\mathrm{N}}=\mathbf{u}\cdot\mathbf{n}$, $u_{\mathrm{T}}=\mathbf{u}\cdot\mathbf{n}^{\perp}$ with the other variables defined as before. Since $\mathbf{n}$ is a unit vector, $u^2+v^2=(u^{\mathrm{N}})^2+(u^{\mathrm{T}})^2$ so the formula for the internal energy e can use either quantity. From these equations, it is clear that the eigenvectors and eigenvalues of the linearized system, as well as the solution to the Riemann problem, are given by those for the 1-dimensional gas dynamucs equations, with u^{T} being treated as a passively advected quantity. Hence, we can use the techniques of [4, 7] for calculating solutions to the Riemann problem and for manipulating characteristic variables.

Although the algorithm described here follows the same basic outline as those given in the previous two sections, there are some differences, mainly with the calculation of $\hat{U}_{i+1/2,j,S}$. For the purpose of calculating $\hat{U}_{i+1/2,j,S}$, we make a nonlinear change of variables, performing the difference calculation of (3.5) in terms of the primitive variables ρ, u, v, p, as was done in [7] for gas dynamics in one space variable. We then transform back to the conserved variables to calculate $U^{n+1/2}_{i+1/2,j,S}$. This procedure enables us to perform our central difference calculation componentwise on the primitive variables, using formulas similar to (1.9), rather than on the amplitudes of an expansion of $\Delta^{\xi}U$ in terms of the right eigenvectors. Also, since we are working in terms of the primitive variables, we can use the more

elaborate central difference algorithm given in [6], which gives rise to a steeper representation of dicontinuities than (1.9).

In order to justify the use of the more elaborate algorithm for computing $\partial U/\partial\xi$ and, more generally, to understand the errors introduced by using difference approximations to $\partial U/\partial\xi$, such as (2.5), it is useful to make a local change of variables $(\xi, \eta) \leftrightarrow (a, b)$

$$a(\xi,\eta)=\int_{\xi_i}^{\xi}\left(\left(\frac{\partial x}{\partial\xi}\right)^2+\left(\frac{\partial y}{\partial\xi}\right)^2\right)^{1/2}d\xi'$$
$$b(\xi,\eta)=\int_{\eta_j}^{\eta}\left(\left(\frac{\partial x}{\partial\eta}\right)^2+\left(\frac{\partial y}{\partial\eta}\right)^2\right)^{1/2}d\eta'. \tag{4.3}$$

The coordinate (a, b) measure arc length along the grid lines $\{\eta = \text{const}\}$, $\{\xi = \text{const}\}$, respectively. It is easy to check that, for (ξ, η) sufficiently close to (ξ_i, η_j) the Jacobian of the above map is nonsingular, since the cross derivatives $\partial a/\partial\eta$, $\partial b/\partial\xi = O((\xi-\xi_i), (\eta-\eta_j))$. Using the chain rule, we compute $\partial U/\partial\xi$ to be

$$\frac{\partial U}{\partial\xi}\Delta\xi=\frac{\partial U}{\partial a}\frac{\partial a}{\partial\xi}\Delta\xi+\frac{\partial U}{\partial b}\frac{\partial b}{\partial\xi}\Delta\xi.$$

Thus, the central difference approximation to $\partial U/\partial\xi$ used in (3.8) can be viewed as using a central difference approximation for $\partial U/\partial a$ and dropping the term proportional to $\partial b/\partial\xi$, since it is of one order smaller in the mesh spacing. In terms of the mesh in physical space, this corresponds to the assumption that the arc length along each of the coordinate directions is a smoothly varying function of the other coordinate. This is a condition satisfied in a wide variety of applications, even when the grid mapping as a whole is not smooth, such as in the case of highly stretched grids used in aerodynamics calculations. In the latter situation, one can retain the formalism developed here but use an approximation to the derivatives appropriate for a strongly varying mesh in the a- or b- direction.

In terms of the coordinate system (4.3), we can express $U^{n+1/2}_{i+1/2,j,S}$ in the form

$$\hat{U}_{i+1/2,j,S}=U^n_{i,j}+\left(\pm\frac{1}{2}-\frac{\Delta t\,\Delta b_{i+k,j}}{2\sigma_{i+k,j}}\mathbf{n}^b_{i+k,j}\cdot A(U^n_{i+k,j})\right)\frac{\partial U}{\partial a}\Delta a_{i,j} \tag{4.4}$$

$$\begin{aligned}U^{n+1/2}_{i+1/2,j,S}=\hat{U}_{i+1/2,j,S}-\frac{\Delta t}{2\sigma_{i+k,j}}\,[&\Delta a_{i+k,j+1/2}\mathbf{n}^a_{i+k,j+1/2}\mathbf{F}(U^T_{i+k,j+1/2})\\&-\Delta a_{i+k,j-1/2}\mathbf{n}^a_{i+k,j-1/2}\cdot\mathbf{F}(U^T_{i+k,j-1/2})\\&+\mathbf{F}(U^n_{i+k,j})\cdot(\Delta b_{i+1/2+k}\mathbf{n}^b_{i+1/2+k,j}-\Delta b_{i-1/2+k}\mathbf{n}^b_{i-1/2+k,j})],\end{aligned} \tag{4.5}$$

where

$$
\begin{aligned}
(\Delta a)_{i,j[+1/2]} &= ((\Delta^{\xi} x)^2_{i,j[+1/2]} + (\Delta^{\xi} y)^2_{i,j[+1/2]})^{1/2} \\
(\Delta b)_{i[+1/2],j} &= ((\Delta^{\eta} x)^2_{i[+1/2],j} + (\Delta^{\eta} y)^2_{i[+1/2],j})^{1/2} \\
\mathbf{n}^b_{i[+1/2],j} &= \frac{(\Delta^{\eta}\mathbf{x})^{\perp}_{i[+1/2],j}}{\Delta b_{i[+1/2],j}} \\
\mathbf{n}^a_{i,j[+1/2]} &= -\frac{(\Delta^{\xi}\mathbf{x})^{\perp}_{i,j[+1/2]}}{\Delta a_{i,j[+1/2]}}.
\end{aligned}
\tag{4.6}
$$

We calculate $\hat{U}_{i+1/2,j,S}$ by transforming to the variables $V=(\rho, u, v, p)^t$ before applying (4.5):

$$
\begin{aligned}
V^n_{i,j} &= V(U^n_{i,j}) \\
\hat{V}_{i+1/2,j,S} &= \tilde{V}_S + P_S(V^n_{i,j} - \tilde{V}_S) + P_S\left(\pm\frac{1}{2} - \frac{\Delta t\, \Delta b_{i+k,j}}{2\sigma_{i+k,j}} T^{-1}_{i+k,j} A^a_{i+k,j} T_{i+k,j}\right) \\
&\quad \times \frac{\partial V}{\partial a} \Delta a_{i+k,j} \\
\hat{U}_{i+1/2,j,S} &= U(\hat{V}_{i+1/2,j,S}).
\end{aligned}
\tag{4.7}
$$

Here $T_{i,j} = \nabla_V U|_{U^n_{i,j}}$ and P_S is defined by $P_S w = \sum_{\nu:\pm\lambda^{a,\nu}_{i+k,j}>0} (l^{a,\nu}_{i+k,j} w)\, r^{a\nu}_{i+k,j}$, where $l^{a,\nu}_{i,j}$, $r^{a,\nu}_{i,j}$, $\lambda^{a,\nu}_{i,j}$, $\nu = 1, \ldots, 4$ are the eigenvectors and eigenvalues of $T^{-1}_{i,j} \cdot A^a_{i,j} \cdot T_{i,j}$:

$$
\lambda^{a,1} = \mathbf{u}\cdot\mathbf{n}^b - c, \qquad \lambda^{a,2} = \lambda^{a,3} = \mathbf{u}\cdot\mathbf{n}^b, \qquad \lambda^{a,4} = \mathbf{u}\cdot\mathbf{n}^b + c
$$

$$
r^{a,1} = \begin{pmatrix} 1 \\ -\dfrac{n^b_x c}{\rho} \\ -\dfrac{n^b_y c}{\rho} \\ c^2 \end{pmatrix}, \qquad r^{a,2} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad r^{a,3} = \begin{pmatrix} 0 \\ -n^b_y \\ b^b_x \\ 0 \end{pmatrix}, \qquad r^{a,4} = \begin{pmatrix} 1 \\ \dfrac{n^b_x c}{\rho} \\ \dfrac{n^b_y c}{\rho} \\ c^2 \end{pmatrix}
$$

$$
\begin{aligned}
l^{a,1} &= \left(0, -\frac{n^b_x \rho}{2c}, -\frac{n^b_y \rho}{2c}, \frac{1}{2c^2}\right) \\
l^{a,2} &= \left(1, 0, 0, -\frac{1}{c^2}\right) \\
l^{a,3} &= (0, -n^b_y, n^b_x, 0) \\
l^{a,4} &= \left(0, \frac{n^b_x \rho}{2c}, \frac{n^b_y \rho}{2c}, \frac{1}{2c^2}\right).
\end{aligned}
$$

Here $\mathbf{n}^b = (n_x^b, n_y^b)$ and the subscripts i, j are suppressed. The time step control (3.12) in terms of the above eigenvalues, is given by

$$\max_{i,j,\nu}\left(\left|\lambda_{i,j}^{a,\nu}\frac{\Delta b_{i,j}\,\Delta t}{\sigma_{i,j}}\right|, \left|\lambda_{i,j}^{b,\nu}\frac{\Delta a\, i,j\,\Delta t}{\sigma_{i,j}}\right|\right) \leqslant 1.$$

The approximation to $(\partial V/\partial a)|_{i,j}\,\Delta a_{i,j}$ we use is obtained by using a formula like (1.9) for each component of V. For example, we define, for $q = p, \rho, u, v$,

$$\begin{aligned}(\Delta_{\lim}^a q)_{i,j} &= 2\min(|q_{i+1,j}^n - q_{i,j}^n|, |q_{i,j}^n - q_{i-1,j}^n|)\\ &\qquad \text{if} \quad (q_{i+1,j}^n - q_{i,j}^n)(q_{i,j}^n - q_{i-1,j}^n) > 0,\\ &= 0 \qquad \text{otherwise},\\ (\Delta_f^a q)_{i,j} &= \min(\tfrac{1}{2}|q_{i+1,j}^n - q_{i+1,j}^n|, (\Delta_{\lim}^a q)_{i,j}) \times \operatorname{sgn}(q_{i+1,j}^n - q_{i+1,j}^n))\end{aligned}$$

and set $(\Delta^a q)_{i,j} = (\Delta_f^a q)_{i,j}$ to obtain the algorithm analogous to (1.9). In the calculations presented in Section 5, we use the following algorithm, taken from [5], which yields a steeper representation of discontinuities:

$$(\Delta^a q)_{i,j} = \min\left(4\,\frac{|q_{i+1,j} - q_{i-1,j} - (1/4)((\Delta_f^a q)_{i+1,j} + (\Delta_f^a q)_{i-1,j})|\,\Delta a_{i,j}}{(\Delta a_{i-1,j} + 4\,\Delta a_{i,j} + \Delta a_{i+1,j})}, (\Delta_{\lim}^a q)_{i,j}\right)$$
$$\times \operatorname{sgn}(q_{i+1,j}^n - q_{i-1,j}^n).$$

Given the values for $\Delta^a V$, we can give explicit formulas for $\hat{V}_{i+1/2,j,S}$:

$$\tilde{V}_L = V_{i,j}^n + \left(\frac{1}{2} - \max(\mathbf{u}_{i,j}^n \cdot \mathbf{n}_{i,j}^b + c_{i,j}^n, 0)\,\frac{\Delta t\,\Delta b_{i,j}}{2\sigma_{i,j}}\right)\Delta^a V_{i,j}$$

$$\tilde{V}_R = V_{i+1,j}^n - \left(\frac{1}{2} + \min(\mathbf{u}_{i+1,j}^n \cdot \mathbf{n}_{i+1,j}^b - c_{i+1,j}, 0)\,\frac{\Delta t\,\Delta b_{i+1,j}}{2\sigma_{i+1,j}}\right)\Delta^a V_{i+1,j}$$

$$\hat{V}_{i+1/2,j,S} = \tilde{V}_S + \sum_{\nu} \beta_{i+1/2,j,S}^{\nu} r_{i+k,j}^{a,\nu}$$

$$\begin{aligned}\beta_{i+1/2,j,L}^{\nu} &= \frac{\Delta t\,\Delta b_{i,j}}{2\sigma_{i,j}}(\lambda_{i,j}^{a,4} - \lambda_{i,j}^{a,\nu})((l_{i,j}^{a,\nu} \cdot \Delta^a V_{i,j}) && \text{if} \quad \lambda_{i,j}^{a,\nu} > 0,\\ &= 0 && \text{otherwise:}\\ \beta_{i+1/2,j,R}^{\nu} &= \frac{\Delta t\,\Delta b_{i+1,j}}{2\sigma_{i+1,j}}(\lambda_{i+1,j}^{a,1} - \lambda_{i+1,j}^{a,\nu})(l_{i+1,j}^{a,\nu} \cdot \Delta^a V_{i+1,j}) && \text{if} \quad \lambda_{i+1,j}^{a,\nu} < 0,\\ &= 0 && \text{otherwise.}\end{aligned}$$

The formulas for $\hat{V}_{i,j+1/2,S}$ are identical to those given above, with the interchange of i and j, $\mathbf{n}^a$ and $\mathbf{n}^b$.

The calculation of $U^{n+1/2}_{i+1/2,j,S}$ given $\hat{U}_{i+1/2,j,S}$ is given by (4.5), with $U^{\mathrm{T}}_{i,j+1/2}$ the solution to the Riemann problem for the equations projected in the $\mathbf{n}^a_{i,j+1/2}$ direction, with left and right states given by $U^{\mathrm{T}}_{i,j+1/2,S} = \hat{U}_{i,j+1/2,S}$ or $U^{\mathrm{T}}_{i,j+1,2,s} = U^n_{i,j+k}$. In the calculations shown below, we use the latter choice.

The final conservative difference step is given by (3.3). We define

$$F^{\xi}_{i+1/2,j} = \begin{pmatrix} m^{n+1/2}_{i+1/2,j} \\ m^{n+1/2}_{i+1/2,j} u^{n+1/2}_{i+1/2,j} + p^{n+1/2}_{i+1/2,j}\, \Delta^{\eta} y_{i+1/2,j} \\ m^{n+1/2}_{i+1/2,j} v^{n+1/2}_{i+1/2,j} - p^{n+1/2}_{i+1/2,j}\, \Delta^{\eta} x_{i+1/2,j} \\ m^{n+1,2}_{i+1/2,j} (E^{n+1/2}_{i+1,2,j} + p^{n+1/2}_{i+1/2,j}/\rho^{n+1/2}_{i+1/2,j}) \end{pmatrix}$$

$$F^{\eta}_{i,j+1/2} = \begin{pmatrix} m^{n+1/2}_{i,j+1/2} \\ m^{n+1/2}_{i,j+1/2} u^{n+1/2}_{i,j+1/2} - p^{n+1/2}_{i,j+1/2}\, \Delta^{\xi} y_{i,j+1/2} \\ m^{n+1/2}_{i,j+1,2} v^{n+1/2}_{i,j+1/2} p^{n+1/2}_{i,j+1/2}\, \Delta^{\xi} x_{i,j+1/2} \\ m^{n+1/2}_{i,j+1/2} (E^{n+1/2}_{i,j+1/2} + p^{n+1/2}_{i,j+1,2}/\rho^{n+1/2}_{i,j+1/2}) \end{pmatrix},$$

where $m^{n+1/2}_{i+1/2,j} = \Delta b_{i+1/2,j} \rho^{n+1/2}_{i+1/2,j} (\mathbf{n}^b_{i+1/2,j} \cdot \mathbf{u}^{n+1/2}_{i+1/2,j})$, $m^{n+1/2}_{i,j+1/2} = \Delta a_{i,j+1/2} \rho^{n+1/2}_{i,j+1/2} (\mathbf{n}^a_{i,j+1/2} \cdot \mathbf{u}^{n+1/2}_{i,j+1/2})$ are the mass fluxes through the zone edges at $(i+\frac{1}{2}, j)$ and $(i, j+\frac{1}{2})$. Then (3.3) is given by

$$U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\sigma_{i,j}} (F^{\xi}_{i-1/2,j} - F^{\xi}_{i+1,2,j} + F^{\eta}_{i,j-1/2} - F^{\eta}_{i,j+1/2}).$$

Dissipation Mechanisms

In [7], it was noticed that, in one space dimension, and near strongly nonlinear shocks, the dissipation implicit in monotonicity constraints such as (3.6) and (4.8), was insufficient to guarantee the correct jump in the Riemann invariants transported along the characteristic families which cross the shock. For that reason, it was suggested that additional dissipation be added to the algorithm near such discontinuities in the form of flattening of the interpolation functions and by adding a small viscous dissipation term to the fluxes. Since both these forms of dissipation were required for 1-dimensional problems, it is expected that similar dissipation would be required for the present algorithm, since, for 1-dimensional problems, it is similar to the algorithm in [7]. The second-order artificial viscosity used in [7] can be applied without modification to the present algorithms simply by adding the dissipative flux to each of the four fluxes, prior to the conservative differencing step. The form these dissipative fluxes take in the case of a general quadrilateral grid is also standard; see, e.g., [19]. The simplest flattening algorithm in [7] can be used, with one important modification: in each zone, the slopes corresponding to the

derivatives in each of the grid directions should be flattened by the same amount. We define flattening χ^a, χ^b,

$$\tilde{\chi}^a_{i,j} = \zeta\left(\frac{|p_{i+1,j} - p_{i-1,j}|}{|p_{i+2,j} - p_{i-2,j}|}\right) \quad \text{if } (\mathbf{u}_{i-1,j} - \mathbf{u}_{i+1,j}) \cdot \mathbf{n}^a_{i,j} > 0, \frac{|p_{i+1,j} - p_{i-1,j}|}{\min(p_{i+1,j}, p_{i-1,j})} > \delta$$
$$= 0 \qquad \text{otherwise} \tag{4.9}$$
$$\chi^a_{i,j} = \min(\tilde{\chi}^a_{i-s_{ij},j}, \tilde{\chi}^a_{i,j}),$$

where

$$\zeta(z) = 0 \qquad \text{if} \quad z > z_1,$$
$$= 1 \qquad \text{if} \quad z < z_0,$$
$$= 1 - \frac{z - z_0}{z_1 - z_0} \qquad \text{if} \quad z_0 < z < z_1,$$

and

$$s_{i,j} = \text{sign}(p_{i+1,j} - p_{i-1,j}).$$

We define $\chi^b_{i,j}$ similarly, with the roles of i and j reversed. Then the slopes $\Delta^a q$, $\Delta^b q$ obtained from (4.8) are reset to

$$\Delta^a q_{i,j}, \Delta^b q_{i,j} \to \chi_{i,j} \Delta^a q_{ij}, \chi_{i,j} \Delta^b q_{i,j}, \tag{4.10}$$

where

$$\chi_{i,j} = \min(\chi^a_{i,j}, \chi^b_{i,j}).$$

In the runs discussed in the next section, the parameters in the above algorithm were set to be $\delta = 0.33$, $z_0 = 0.75$, $z_1 = 0.85$. In addition, we used the 2-dimensional Lapidus viscous flux discussed in [7] with a coefficient of 0.1. These were the choice of the parameters used in the corresponding algorithms for operator split calculations described in [7] and have been found to give adequate results when used with the present algorithm over a wide range of problems.

Boundary Conditions

It is straightforward to impose various continuation-type boundary conditions (inflow, outflow, periodic, etc.) in regions where the grid has a natural extension beyond the computational domain. Since the numerical domain of dependence of a grid point is contained in the 9×9 block of grid points containing the point at the center, then one can extend the original computational mesh by four grid points in each direction and set the values of the extended part of the grid at the beginning of each time step using the boundary conditions, thus supplying sufficient data to calculate the values on the original grid.

The most common situation where one cannot extend the grid is in the case of an impermeable surface, particularly on a body-fitted grid. Let us assume, for example, that the curve $\{\xi(\mathbf{x}) = \xi_{i_0-1/2}\}$ is a reflecting surface, with the fluid contained in the region $\{\xi(\mathbf{x}) > \xi_{i_0-1/2}\}$. The algorithm described above can be applied without modification, if we specify values for the slopes $\Delta_f^b q_{i_0-1,2,j}$, $\Delta^b q_{i_0-1/2,j}$ and for the fluxes $\mathbf{F}(U^{\mathrm{T}}_{i_0-1/2,j})$, $\mathbf{F}(U^{n+1/2}_{i_0-1\,2,j})$. The slopes are given by

$$\Delta q_{i_0,j} = \Delta_l q_{i_0,j} = 0, \qquad q = p, \rho, \mathbf{n}^{b\perp}_{i_0-1\,2,j}\cdot\mathbf{u}$$

$$\mathbf{n}^b_{i_0-1/2,j}\cdot\Delta_f^b\mathbf{u}_{i_0,j} = \min(|\mathbf{u}_{i_0,j}\cdot\mathbf{n}^b_{i_0-1/2,j}|, 2|(\mathbf{u}_{i_0+1,j}-\mathbf{u}_{i_0,j})\cdot\mathbf{n}^b_{i_0-1/2,j}|)\,\mathrm{sgn}(\mathbf{u}_{i_0,j}\cdot\mathbf{n}^b_{i_0-1/2,j})$$
$$\text{if} \quad (\mathbf{u}_{i_0,j}\cdot\mathbf{n}^b_{i_0-1/2,j})(\mathbf{u}_{i_0+1,j}-\mathbf{u}_{i_0,j})\cdot\mathbf{n}^b_{i_0-1,2,j} > 0$$
$$= 0 \qquad \text{otherwise.} \tag{4.11}$$

Given the slope information, it is possible to calculate $\hat{U}_{i_0-1/2,j,R}$, $U^{n+1/2}_{i_0-1/2,j,R}$. To obtain the states $U^{\mathrm{T}}_{i_0-1/2,j}$, $U^{n+1/2}_{i_0-1/2,j}$, we solve Riemann problems projected in the $\mathbf{n}^b_{i_0-1/2,j}$ direction, with left and right state given by

$$\hat{q}_{i_0-1/2,j,L}, q^{n+1/2}_{i_0-1/2,j,L} = \hat{q}_{i_0-1/2,j,R}, q^{n+1/2}_{i_0-1/2,j,R}, \qquad q = p, \rho, \mathbf{n}^{b\perp}_{i_0-1,2,j}\cdot\mathbf{u}$$
$$\mathbf{n}^b_{i_0-1,2,j}\cdot\hat{\mathbf{u}}_{i_0-1/2,j,L}, \mathbf{n}^b_{i_0-1,2,j}\cdot\mathbf{u}^{n+1/2}_{i_0-1/2,j,L} = -\mathbf{n}^b_{i_0-1/2,j}\cdot\hat{\mathbf{u}}_{i_0-1/2,j,R}, -\mathbf{n}^b_{i_0-1,2,j,R}. \tag{4.12}$$

With this choice of left and right states, it is clear that $\mathbf{u}_{i_0-1/2,j} = 0$, so that the advective terms in the fluxes at $(i_0-\frac{1}{2}, j)$ vanish, leaving only the pressure terms in the x- and y-momentum equations. Whatever approximate solution to the Riemann problem is used should guarantee that the advective terms vanish in the flux calculation at the wall.

5. Numerical Results

The gas dynamics algorithm described here has been used in a variety of applications in two dimensions, including flow in cascades and channels with body-fitted meshes [9], in adaptive mesh refinement calculations [2], and in a conservative front-tracking algorithm [3]. In addition, various forms of the algorithm for scalar equations have been used to calculate flow in porous media [15].

We will present here two gas dynamics calculations, both done on rectangular grids. The first is the calculation of a steady state regular shock reflection described in [23], which has been used extensively as a test problem for numerical methods used in aerodynamic calculations [25]. The second test problem is the double Mach reflection of a shock off an oblique surface, used in [22] as a test problem for comparing the performance of various difference methods on problems involving strong shocks. Since our purpose is to demonstrate that the current method has the same resolution as the corresponding operator split algorithm, we present also a calculation of the latter problem performed by using in an operator

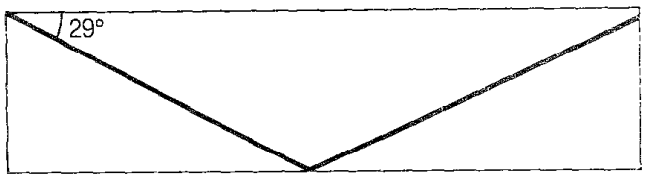


FIG. 5. Steady state regular reflection problem.

split formulation the 1-dimensional algorithm obtained by restricting the algorihtm described in Section 4 to one dimension.

In the first test problem, the computational domain is a rectangle of length 4 and height 1 (Fig. 5). This domain is divided into a 60×20 rectangular grid, with $\Delta x = \frac{1}{15}$, $\Delta y = \frac{1}{20}$. The boundary conditions are that of a reflecting surface along the bottom boundary, supersonic outflow along the right boundary, and Dirichlet conditions on the other two sides, given by

$$(\rho, u, v, p)|_{(0, y, t)} = (1., 2.9, 0., 1/1.4)$$

$$(\rho, u, v, p)|_{(x, 1, t)} = (1.69997, 2.61934, .50632, 1.52819).$$

Initially, we set the solution in the entire domain to be that at the left boundary; we then iterate for 500 time steps using a CFL condition of 0.9, at which time the solution reaches a steady state.

In Fig. 6, we show a contour plot of the pressure. The contours are equally

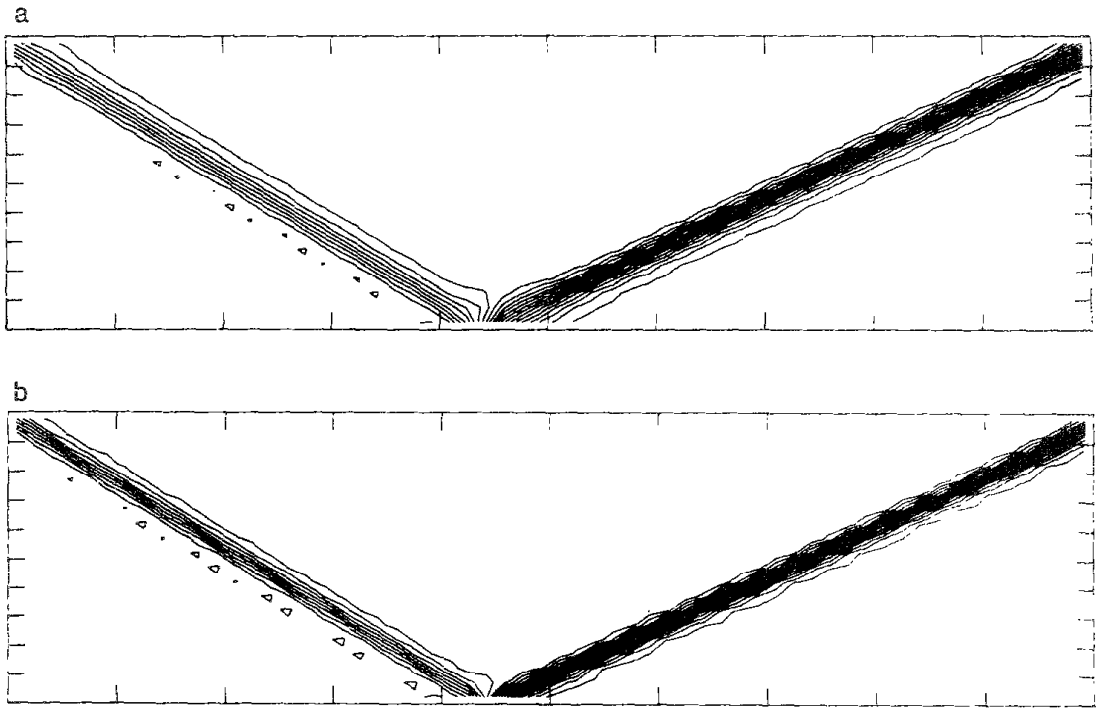


FIG. 6. Numerical solution to regular reflection problem: (a) with flattening; (b) without flattening.

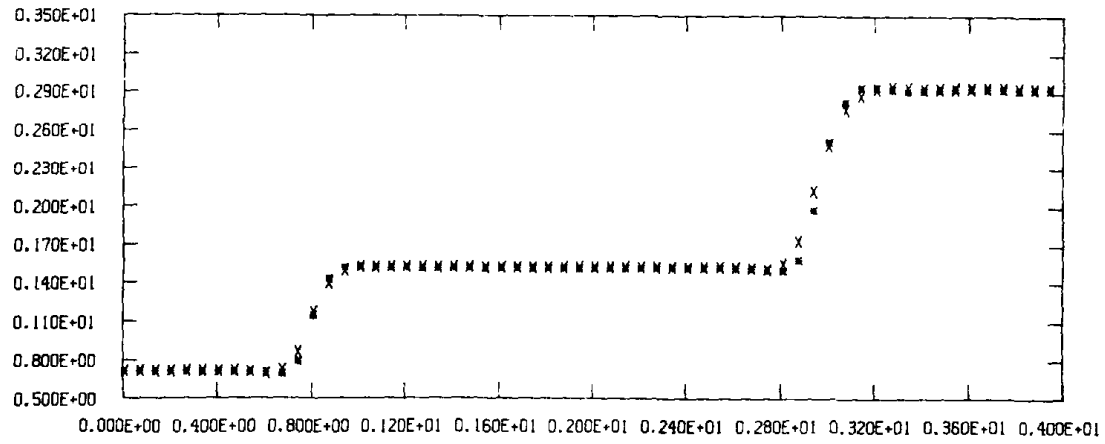


FIG. 7. Comparison of pressure profiles for regular reflection problem along the line $y = 0.525$ ($j = 11$): x—with flattening, $*$—without flattening.

spaced, with contour levels of 0.1, beginning at 0. The shocks have a nearly monotone transition, and are fairly narrow, with some slight spreading on the high pressure side of each shock. This spreading is due to the flattening algorithm (4.10). We see this in Fig. 7, where we plot profiles of the solution at $y = 0.525$, computed with and without flattening. The width of the shocks is about 2–$2\frac{1}{2}$ zones in the normal direction, where this figure is obtained by counting the number of points in the transition in Fig. 7, and multiplying it by $\sin(\tan^{-1}((\Delta x/\Delta y)|\tan(\alpha)|))$, where α is the angle between the direction tangent to the shock and the x direction. The shock transition with flattening is slightly broader; however, the transition without flattening has some low-amplitude oscillations, which are not present in the solution obtained with flattening. Even though the shocks are supersonic on both

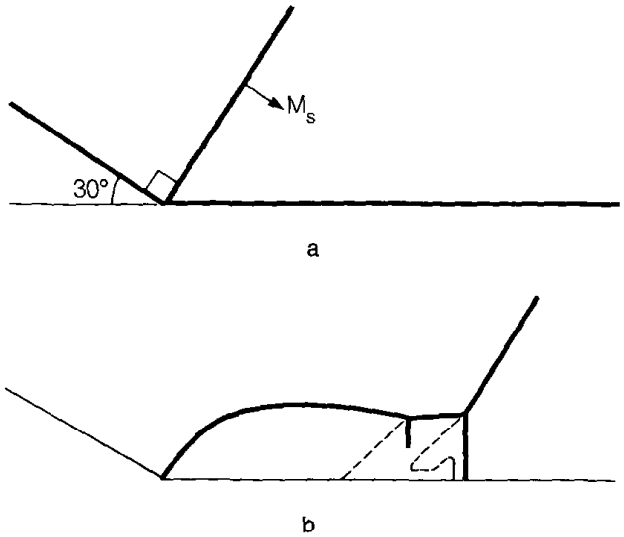


FIG. 8. Ramp reflection problem: (a) initial configuration; (b) double Mach reflection at later times: solid lines are shocks; dotted lines are slip surfaces.

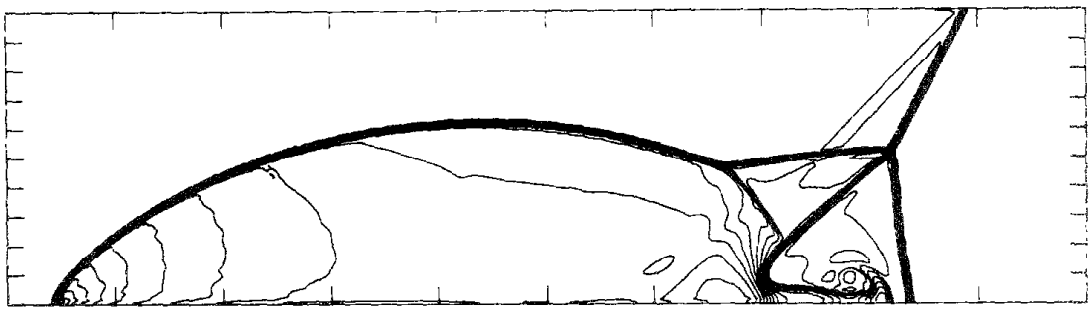


FIG. 9. Numerical solution of ramp problem using the method described in Section 4. The mesh is a rectangular mesh of 400×100 zones, with the reflecting wall beginning 20 mesh lengths from the lower left corner. $\Delta x = \Delta y = \frac{1}{120}$, and the time shown is $t = 0.2$; thus this calculation corresponds to the finest grid results in [22]

sides, there is no difficulty with uncontrolled diffusion of the discontinuities. This is in contrast to the results obtained with first-order upwind methods, where steady shocks remain quite sharp if the transition is supersonic/subsonic, but which spread over many zones if the transition is supersonic/supersonic. Indeed, the main difficulty for the present method is to ensure that the shocks are broad enough so that sufficient dissipation occurs across the shock, as was the case with the operator split second-order methods.

The second test problem is unsteady shock reflection problem. A planar shock is incident on an oblique surface, with the surface at a 30° angle to the direction of propagation of the shock (Fig. 8). The fluid in front of the shock has zero velocity, and the shock Mach number is equal to 10. The solution to this problem is self-similar, with U a function of (x, y, t) only in the combination $(x/t, y/t)$. In Fig. 9, we show the results of calculation of this test problem performed with the present unsplit second-order method; in Fig. 10, the corresponding results obtained with the operator split method. The results of the two calculations are essentially identical, supporting the assertion that the unsplit method has the same resolution as the corresponding operator split method. However, a considerable degree of care was required in the unsplit scheme for this to be the case. The choice of (2.15), rather

FIG. 10. Numerical solution of ramp reflection problem, using operator split method, with numerical parameters the same as for Fig. 9.

than (2.13), in calculating the transverse derivative in the predictor step is essential; otherwise, one obtains considerably lower resolution in the jet along the wall in the double Mach region. The accuracy in the double Mach region is also sensitive to the reflecting boundary conditions. The former difficulty has no analogue in the operator split method; as for the latter problem, the operator split method gives the same results which much simpler boundary conditions. Finally, the multidimensional flattening algorithm given by (4.10) was required to eliminate low-amplitude noise behind the shocks, whereas the operator split algorithms required only the 1-dimensional flattening algorithm in [7] to be applied in each sweep.

6. Discussion and Conclusions

In this paper, we have derived explicit second-order Godunov-type methods in two space variables by using the wave propagation properties for multidimensional hyperbolic equations and by limiting some of the second-order terms to suppress oscillations. The calculations in Section 5 indicate that we have been successful in the goal stated in the Introduction of producing an algorithm with comparable performance to the operator split second-order Godunov methods, at a comparable cost. In retrospect, this is not surprising, since the multidimensional algorithm consists of combinations of the 1-dimensional operators which appear in the operator split schemes. In particular, the same Riemann problems appear in the present method as in the operator split methods, since in the former case averaging the solution to the characteristic form of the equations over a zone edge provides, via (2.4), a natural choice of a direction in which to project the multidimensional equations for solving the Riemann problem. However, there are differences between the present algorithms and the operator split approach. The algorithms discussed here are somewhat more expensive, requiring twice as many solutions to the Riemann problem as the corresponding operator split algorithm. Since the cost of solving the Riemann problem for a polytropic equations of state constitutes half the cost of the calculation in one dimension [6], this leads to an algorithm which takes 50% more time than the operator split algorithm. In the regular reflection problem, the vectorized implementation on the Cray 1 advanced about 24,000 zones by one time step in each cpu second, consistent with this estimate and the timing figures for the corresponding 1-dimensional algorithm given in [6]. Also, the multidimensional algorithms appear to be more sensitive to various details of the implementation, requiring a greater degree of care, such as for the reflecting boundary conditions (4.11)–(4.12), and for the flattening algorithm (4.10).

There are a number of straightforward applications and extensions of the methods described here. It is possible to introduce quadratic interpolants, as in [7], to evaluate $\hat{U}$ in the predictor step in order to improve the resolution of linear discontinuities by means of contact detection and steepening. Conservation laws for which the fluxes have an explicit spatial dependence, such as for incompressible multiphase flow in porous media, can be easily treated using similar techniques to

the ones used for the general quadrilateral meshes. The treatment of a general equation of state via the techniques in [6] is accomplished by introducing an additional transport equation for $\gamma = p/\rho e + 1$ for use in the predictor step for the transverse derivatives. Thus introduces some additional complication into the method, which is more than offset by the fact one need only evaluate the equation of state once per zone per time step.

There are some problems for which the formalism given here is attractive, but for which the extensions are not entirely straightforward. One of these is the extension of this method for calculation of problems in Lagrangian coordinates in two dimensions. The difficulty here is that the motion of the grid must be obtained from the solution itself; unlike in one dimension, neither the solution nor the fluxes are defined at the corners of the mesh, where it is most natural to specify the motion of the grid. Consequently, some form of averaging of the velocities must be introduced in order to move the grid, but one which does not degrade the resolution of the method [17]. Finally, there is the question of the extension of these ideas to three dimensions. If we just take as our advection algorithm the 3-dimensional analogue of (1.2), we arrive at an algorithm for systems which satisfies the properties (1)–(3) in the Introduction, but requires 12 solutions to the Riemann problem per zone per time step; this is in contrast to the 3 solutions required by an operator split method. The large number of solutions to the Riemann problem comes from the fact that for each coordinate direction in three dimensions, the analogue of the predictor step for the transverse derivatives (2.9) requires a calculation comparable to the full 2-dimensional calculation described in this paper. However, if we are willing to relax the third requirement somewhat, we obtain an algorithm which requires only 6 solutions to the Riemann problem by using the extension of donor-cell differencing to systems to evaluate the transverse derivatives in the predictor step; equivalently, we would be ignoring the contributions due to transport from zones offset by one mesh length in all three directions, which correspond to third-order terms in the truncation error. In both cases, we would obtain algorithms which, for 2-dimensional problems aligned with one of the mesh directions, give identical results to the algorithms described in this paper. The question as to what the appropriate formulation is for problems in three dimensions is undoubtedly problem dependent, and probably can be resolved only by numerical experiments.

REFERENCES

1. *Methods of Computational Physics*, Vol. 3, edited by B. Alder and S. Fernbach (Academic Press, New York, 1964).
2. M. BERGER AND P. COLELLA, Lawrence Livermore National Laboratory Report UCRL-97196; *J. Comput. Phys.* **82**, 64 (1989).
3. I.-L. CHERN AND P. COLELLA, Lawrence Livermore National Laboratory Report UCRL-97200, *J. Comput. Phys.*, in press.
4. P. COLELLA, *SIAM J. Sci. Stat. Comput.* **3**, 76 (1982).
5. P. COLELLA, *SIAM J. Sci. Stat. Comput.* **6**, 107 (1985).

6. P. COLELLA AND H. M. GLAZ, *J. Comput. Phys.* **59**, 264 (1985).
7. P. COLELLA AND P. R. WOODWARD, *J. Comput. Phys.* **54**, 174 (1984).
8. R. COURANT AND D. HILBERT, *Methods of Mathematical Physics*, Vol. II (Interscience, New York, 1963).
9. S. EIDELMAN, P. COLELLA, AND R. P. SHREEVE, *AIAA J.* **22**, 1609 (1984).
10. B. A. FRYXELL, P. R. WOODWARD, P. COLELLA, AND K.-H. WINKLER, *J. Comput. Phys.* **63**, 283 (1986).
11. S. K. GODUNOV, A. V. ZABRODYN, AND G. P. PROKOPOV, *USSR Comput. Math. Math. Phys.* **1**, 1187 (1961).
12. J. B. GOODMAN AND R. LEVEQUE, *Math. Comput.* **45**, 15 (1985).
13. A. HARTEN, *J. Comput. Phys.* **49**, 357 (1983).
14. A. HARTEN, "On Second Order Accurate Godunov-type Schemes," 1982 (unpublished).
15. C. H. LAI, G. S. BODVARSSON, AND P. A. WITHERSPOON, "Numerical Studies of Silica Precipitation/Dissolution," Lawrence Berkeley Laboratory Earth Sciences Division, 1985 (unpublished).
16. R. D. RICHTMYER AND K. W. MORTON, *Finite Difference Methods for Initial-Value Problems* (Interscience, New York, 1967).
17. J. S. SALTZMAN AND P. COLELLA, Los Alamos National Laboratory Report LAUR-85-678, 1985 (unpublished).
18. G. R. SHUBIN AND J. B. BELL, *Comput. Meth. Appl. Mech. Eng.* **47**, 47 (1984).
19. J. T. STEGER, *AIAA J.* **16**, 679 (1978).
20. B. VAN LEER, *J. Comput. Phys.* **23**, 276 (1977).
21. B. VAN LEER, in *Computing Methods in Applied Sciences and Engineering VI*, edited by R. Glowinski and J.-L. Lions (North-Holland, Amsterdam, 1984), p. 493.
22. P. R. WOODWARD AND P. COLELLA, *J. Comput. Phys.* **54**, 115 (1984).
23. H. C. YEE, R. F. WARMING, AND A. HARTEN, in *Proceedings, 8th International Conference on Numerical methods in Fluid Dynamics*, Lecture Notes in Physics Vol. 141 (Springer-Verlag, New York/Berlin, 1982), p. 547.
24. S. T. ZALESAK, *J. Comput. Phys.* **31**, 335 (1979).
25. "Proceedings, Sixth AIAA Computational Fluid Dynamics Conference, Danvers, MA, June, 1983."

Appendix A

Reviews on PDEs

1. Properties of PDEs

In this chapter, we study the key defining properties of partial differential equations (PDEs). First of all, there are more than one 'independent' variables $t, x, y, z, ...$ Associated to these is so called a 'dependent' variable u (of course there could be more than one dependent variables) which is a function of those independent variables,

$$u = u(x, y, z, t, ...) \tag{A.1}$$

We now provide a bunch of basic definitions and examples on PDEs.

Definition: A PDE is a relation between the independent variables and the dependent variable u via the partial derivatives of u.

Definition: The order of PDE is the highest derivative that appears.

Example: $F(x, y, u, u_x, u_y) = 0$ is the most general form of first-order PDE in two independent variables x and y.

Example: $F(t, x, y, u, u_t, u_{xx}, u_{xy}, u_{yy}) = 0$ is the most general form of second-order PDE in three independent variables t, x and y.

Example: $u_t - u_{xx} = 0$ is a second-order PDE in two independent variables t and x.

Example: $u_{xxxx} + (u_y)^3 = 0$ is a fourth-order PDE in two independent variables x and y.

Definition: $\mathcal{L}$ is called a linear operator if $\mathcal{L}(u+v) = \mathcal{L}u + \mathcal{L}v$ for any functions u and v.

Definition: A PDE $\mathcal{L}u = 0$ is called a linear PDE if $\mathcal{L}$ is a linear derivative operator.

Definition: A PDE $\mathcal{L}u = g$ is called an inhomogeneous linear PDE if $\mathcal{L}$ is a linear derivative operator and if $g \neq 0$ is a given function of the independent variables. If $g = 0$, it is called a homogeneous linear PDE.

Example: The following PDEs are homogeneous linear:
$u_x + u_y = 0$ (transport); $u_x + yu_y = 0$ (transport); $u_{xx} + u_{yy} = 0$ (Laplace's equation)

Example: The following PDEs are homogeneous nonlinear:
$u_x + uu_y = 0$ (shock wave); $u_{tt} + u_{xx} + u^3 = 0$ (wave with interaction); $u_t + uu_x + u_{xxx} = 0$ (dispersive wave);

Example: The following PDEs are inhomogeneous linear:
$cos(xy^2)u_x - y^2u_y = tan(x^2 + y^2)$

2. Well-posedness of PDEs

When solving PDEs, one often encounters a problem that has more than one solution (non-uniqueness) if few auxiliary conditions are imposed. Then the problem is called underdetermined. On the other hand, if too many conditions are given, there may be no solution at all (non-existence) and in this case, the problem is overdetermined.

The well-posedness property of PDEs is therefore required in order for us to enable to solve the given PDE system successfully. Well-posed PDEs of proper initial and boundary conditions follows the following fundamental properties:

1. Existence: There exists at least one solution $u(x, t)$ satisfying all these conditions,
2. Uniqueness: There is at most one solution,
3. Stability: The unique solution $u(x, t)$ depends in a stable manner on the data of the problem. This means that if the data are changed a little, the corresponding solution changes only a little as well.

3. Classifications of Second-order PDEs

PDEs arise in a number of physical phenomena to describe their natures. Some of the most popular types of such problems include fluid flows, heat transfer, solid mechanics and biological processes. These types of equations often fall into one of three types, (i) hyperbolic PDEs that are associated with advection, (ii) parabolic PDEs that are most commonly associated with diffusion, and (iii) elliptic PDEs that most commonly describe steady states of either parabolic or hyperbolic problems.

In reality, not many problems fall simply into *one* of these three types, rather most of them involve combined types, e.g., advection-diffusion problems.

Mathematically, however, we can rather easily determine the type of a general second-order PDEs, which we are going to briefly discuss here.

In general, let's consider the PDE of form with constants a_{11}, a_{12}, and a_{22}, where not all of them are zeros:

$$a_{11}u_{xx} + 2a_{12}u_{xy} + a_{22}u_{yy} + a_1u_x + a_2u_y + a_0u = 0, \tag{A.2}$$

which is a second-order linear equation in two independent variables x and y with six constant coefficients.

Theorem: By a linear transformation of the independent variables, the equation can be reduced to one of three forms:

1. Elliptic PDE: if $a_{12}^2 < a_{11}a_{22}$, it is reducible to

$$u_{xx} + u_{yy} + L.O.T = 0 \tag{A.3}$$

   where $L.O.T$ denotes all the lower order terms (first or zeroth order terms).

2. Hyperbolic PDE: if $a_{12}^2 > a_{11}a_{22}$, it is reducible to

$$u_{xx} - u_{yy} + L.O.T = 0 \tag{A.4}$$

3. Parabolic PDE: if $a_{12}^2 = a_{11}a_{22}$ (the condition for parabolic is in between those of elliptic and hyperbolic), it is reducible to

$$u_{xx} + L.O.T = 0 \tag{A.5}$$

Remark: Notice the similarity between the above classification and the one in analytic geometry. We know from analytic geometry that, given (again assuming constants a_{11}, a_{12}, and a_{22} where they are not all zeros)

$$a_{11}x^2 + 2a_{12}xy + a_{22}y^2 + a_1x + a_2y + a_0 = 0, \tag{A.6}$$

Then Eq. A.6 becomes

1. Ellipsoid if $a_{12}^2 < a_{11}a_{22}$
2. Hyperbola if $a_{12}^2 > a_{11}a_{22}$
3. Parabola if $a_{12}^2 = a_{11}a_{22}$.

Note again that parabola is in between ellipsoid and hyperbola. See Fig. 1 for an illustration.

Example: $u_{xx} - 5u_{xy} = 0$ is hyperbolic; $4u_{xx} - 12u_{xy} + 9u_{yy} + u_y = 0$ is parabolic; $4u_{xx} + 6u_{xy} + 9u_{yy} = 0$ is elliptic.

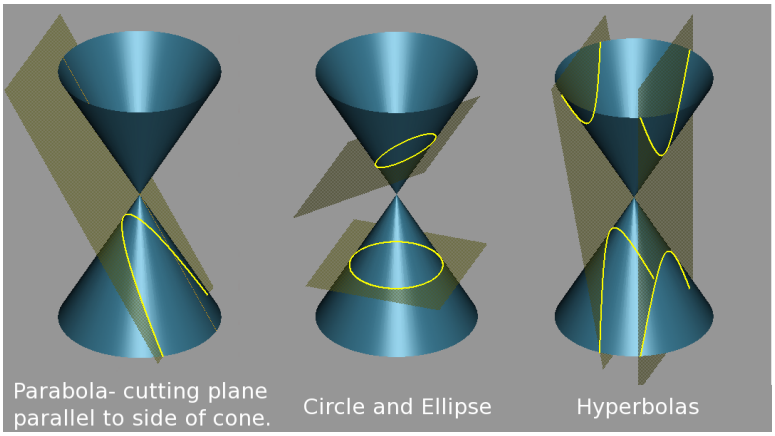


Figure 1. Three major types of conic section from analytic geometry – Image source: Wikipedia

Example: The wave equation is one of the most famous examples in hyperbolic PDEs. We write the wave equation as

$$u_{tt} = c^2 u_{xx} \text{ for } -\infty < x < \infty, c \neq 0. \tag{A.7}$$

Factoring the derivative operator, we get

$$\Big(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\Big)\Big(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\Big)u = 0 \tag{A.8}$$

Considering the characteristic coordinates $\xi = x + ct$ and $\eta = x - ct$, we obtain

$$0 = \Big(\frac{\partial}{\partial t} - c\frac{\partial}{\partial x}\Big)\Big(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\Big)u = \Big(-2c\frac{\partial}{\partial \xi}\Big)\Big(2c\frac{\partial}{\partial \eta}\Big)u \tag{A.9}$$

Hence, we conclude that the general solution must have a form $u(x,t) = f(x + ct) + g(x - ct)$, the sum of two functions, one (g) is a wave of any shape traveling to the the *right* at speed c, and the other (f) with another arbitrary shape traveling to the the *left* at speed c. We call the two families of lines, $x \pm ct = constant$, the characteristic lines of the wave equation.

Example: One very simple and famous example in the parabolic PDEs is so called the diffusion equation

$$u_t = ku_{xx}, \text{ with } k \text{ constant and } (x,t) \in D \times T \tag{A.10}$$

One of the important properties in the diffusion equations is to have the maximum principle. Recall that the maximum principle says if $u(x,t)$ is the solution of Eq. A.10 on $D \times T = [x_{min}, x_{max}] \times [T_0, T_1]$ in space-time, then the maximum value of $u(x,t)$ is assumed only on the initial and domain boundary of $D \times T$. That is, the maximum value only occurs either initially at $t = T_0$ or on the sides

$x = x_{min}$ or $x = x_{max}$.

Remark: The fundamental properties of the two types of PDEs can be briefly compared in the following table. The physical meanings in Table 1 are also illustrated in Fig. 2 and Fig. 3.

Table 1. Comparison of Waves and Diffusions: Fundamental properties of the wave and diffusion equations are summarized.

Property	Waves	Diffusions
(1) speed of propagation	finite ($\leq c$)	∞
(2) singularities for $t > 0$?	transported along characteristics (with speed = c)	lost immediately
(3) well-posed for $t > 0$?	yes	yes (at least for bounded solutions)
(4) well-posed for $t < 0$?	yes	no
(5) maximum principle?	no	yes
(6) behavior as $t \to \infty$	energy is constant so does not decay (i.e., simple advection without diffusion)	decays to zero
(7) information	transported	lost gradually

4. Finite difference scheme for 1D advection

Consider a simple advection equation with constant speed $c > 0$:

$$u_t + cu_x = 0, \text{ with } u(x,0) = sin(x), x \in [0, 2\pi] \tag{A.11}$$

with a periodic boundary condition. In order to discretize the system, we first subdivide both spatial and temporal domains as

$$x_i = i\Delta x \text{ and } t^n = n\Delta t, \tag{A.12}$$

where i and n are integers. $\Delta x > 0$ and $\Delta t > 0$ are respectively, a spatial grid spacing and a time step. Let us denote our discrete data at each (x_i, t^n):

$$u_i^n = u(x_i, t^n) \tag{A.13}$$

The forward difference scheme writes

$$u_x(x,t) = \frac{u(x + \Delta x, t) - u(x,t)}{\Delta x} + O(\Delta x), \tag{A.14}$$

$$u_t(x,t) = \frac{u(x, t + \Delta t) - u(x,t)}{\Delta t} + O(\Delta t). \tag{A.15}$$

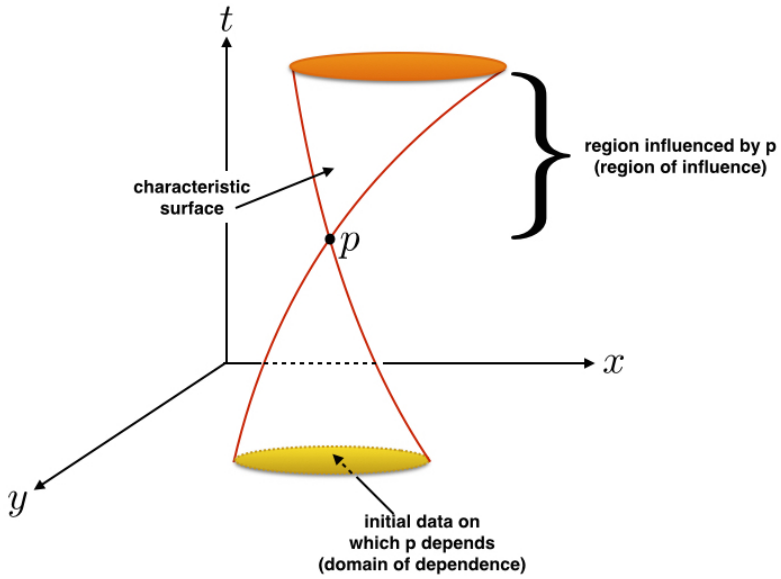


Figure 2. Domain and boundaries for the solution of hyperbolic PDEs in 2D. Note that any information or disturbance introduced at p is going to affect *only* the region called the 'region of influence' but nowhere. Such information is propagated with the finite advection speed along the characteristic surface which forms the conic region of influence. On the other hand, if the characteristic surface can be extended backward in time to the place where the initial data is imposed. This also forms another conic section on the lower part of the figure which is called the 'domain of dependence'.

Dropping the truncation error terms $O(\Delta x)$ and $O(\Delta t)$ yields a simple first-order difference scheme that approximates the advection PDE. As a result, we arrive at a first-order accurate discrete difference equation from an analytic differential equation:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_{i+1}^n - u_i^n}{\Delta x} = 0, \tag{A.16}$$

which gives a temporal update scheme of u_i^{n+1} in terms of the known data at $t = t^n$:

$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}\left(u_{i+1}^n - u_i^n\right) \tag{A.17}$$

On the other hand, if we use a backward difference scheme for u_x

$$u_x(x,t) = \frac{u(x,t) - u(x-\Delta x,t)}{\Delta x} + O(\Delta x), \tag{A.18}$$

we arrive at another first-order difference equation

$$u_i^{n+1} = u_i^n - c\frac{\Delta t}{\Delta x}\left(u_i^n - u_{i-1}^n\right). \tag{A.19}$$

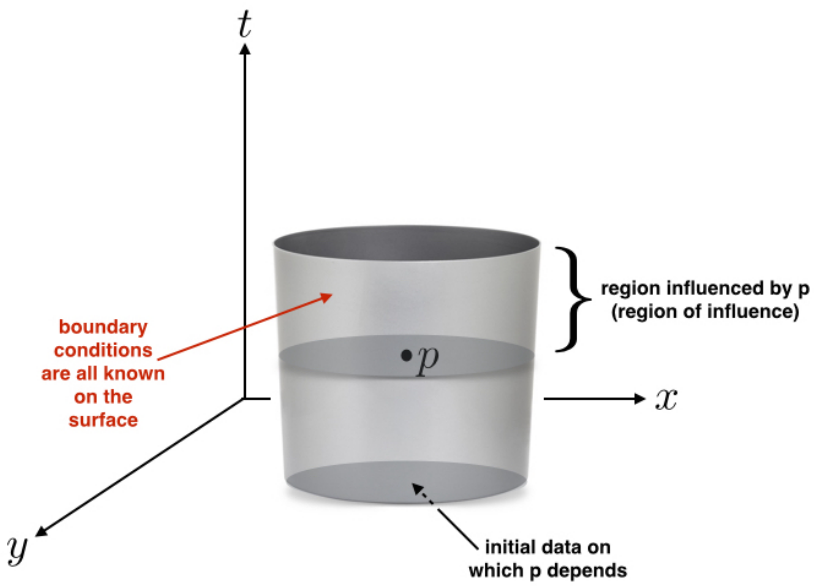


Figure 3. Domain and boundaries for the solution of parabolic PDEs in 2D. Note that from a given point p in the mid plane, there is only one physically meaningful direction that is positive in t. Therefore, any information at p influences the entire region onward from p, called the 'region of influence'. Such information can only marches forward in time under the assumption that all boundary conditions around the surface and the initial condition are known.

Let us choose Δt small enough that

$$|c|\Delta t \leq \Delta x \tag{A.20}$$

Problem 1. Write a simple MATLAB program (or use any other scientific language) in order to numerically solve Eq. A.17 and Eq. A.19. Please make sure your code satisfies the condition in Eq. A.20. Choose $t = t_{max}$ in such that the initial sinusoidal wave makes two complete cycles over the domain (we conveniently assume the cgs unit system – e.g., cm in length, sec in time, $gram$ in mass.).
(a) Use the grid sizes of 16, 32, 64, 128 and 256 and compare your results.
(b) First solve for $c > 0$. Which scheme is better between Eq. A.17 and Eq. A.19?
(c) What happens if $c < 0$?
(d) What happens if your Δt fails to satisfy Eq. A.20 for your choices of c and Δx?
(e) Plot your numerical solutions at $t = t_{cycle1}$ and $t = t_{cycle2}$ on a grid size of 32 using $c > 0$ and the scheme in Eq. A.19. What do you observe?

5. Numerical Solutions of 1D Diffusion

Consider a temporal evolution of solving the classical homogeneous heat equation (or diffusion equation) of the form

$$u_t = \kappa u_{xx} \tag{A.21}$$

with $\kappa > 0$ (Note if $\kappa < 0$ then Eq. A.21 would be a "backward heat equation", which is an ill-posed problem. See Table 1). Along with this equation, let us impose an initial condition at $t = 0$,

$$u(x, 0) = f(x) \tag{A.22}$$

and also the Dirichlet boundary condition on a bounded domain $0 \leq x \leq 1$

$$u(0, t) = g_0(t) \text{ and } u(1, t) = g_1(t), \text{ for } t > 0. \tag{A.23}$$

Use the discretization technique we used in the previous example of the 1D advection finite difference scheme in order to discretize your temporal and spatial domains (i.e., Eq. A.12 and Eq. A.13). As before, we choose the forward difference scheme for temporal discretization as in Eq. A.15. For a spatial discretization, we adopt the standard second-order central difference difference scheme,

$$u_{xx}(x, t) = \frac{u(x + \Delta x, t) - 2u(x, t) + u(x - \Delta x, t)}{\Delta x^2} + O(\Delta x^2), \tag{A.24}$$

which gives a final discrete form of our explicit finite difference scheme for the heat equation:

$$u_i^{n+1} = u_i^n + \kappa \frac{\Delta t}{\Delta x^2} \Big(u_{i+1}^n - 2u_i^n + u_{i-1}^n \Big) \tag{A.25}$$

Similar to the 1D advection case, we choose Δt satisfying

$$\kappa \Delta t \leq \frac{\Delta x^2}{2} \tag{A.26}$$

Problem 2. Write a simple MATLAB program (or use any other scientific language) in order to numerically solve Eq. A.21. The boundary condition is given so as to hold the temperature u to be zero at $x = 0$ and 100° F at $x = 1$ for $t \geq 0$ (i.e., $g_0 = 0$°F and $g_1 = 100$°F.). Your numerical scheme solves three different temporal evolutions for three materials:

(i) iron with $\kappa = 0.230 cm^2/sec$,
(ii) aluminum $\kappa = 0.975 cm^2/sec$, and
(iii) copper with $\kappa = 1.156 cm^2/sec$.

Choose $t = t_{max}$ in each so that each material reaches to a steady state solution. Your initial condition in all three cases is to describe a same initial temperature profile

$$f(x) = 0°\text{F for } 0 \leq x < 1; f(x) = 100°\text{F for } x = 1 \tag{A.27}$$

(a) Use the grid sizes of 16, 32, 64, 128 and 256 and compare your results. What

can you say about the grid resolution study in the diffusion equation as compared to the case of the advection equation?
(b) What happens if your Δt fails to satisfy Eq. A.26 for each κ?
(c) What are your values of t_{max} for three different materials?