Chapter 3: Scalar Advection and Linear Hyperbolic Systems

3.1) Introduction

The previous chapter showed us how we can construct a finite difference approximation (FDA) for any partial differential equation (PDE). Our study of consistency and stability showed us that we can construct FDAs for PDEs and be assured that for well-specified initial and boundary conditions they will converge to the actual solution of the PDE. However, when we applied our ideas to the scalar advection equation we discovered several deficiencies in every linear FDA that we constructed for solving that equation. The deficiencies were especially apparent when we tried to advect discontinuous solutions. The first order accurate advection schemes were too diffusive and dissipative; the second order accurate advection schemes generated spurious extrema at discontinuities. Note though that (at least from a theoretical viewpoint) a discontinuous initial condition is not a well-specified initial condition for a PDE. From a practical viewpoint, we nevertheless do want discontinuities to be advected accurately. In this chapter we start the process of finding a way out of this problem.

We start with the scalar advection equation. We show that there is a pictorial approach to advection that provides several important insights. In particular, we find that we can relate several of the schemes from Section 2.7 to this visually-motivated approach and thereby gain insights into the inner workings of the schemes as well as their deficiencies. Once we understand the source of the deficiencies, we discover a class of second order schemes that minimize diffusion and dissipation while being able to propagate discontinuous solutions without generating spurious overshoots and undershoots. This is done in Section 3.2. In Section 3.3 we formalize our insights from Section 3.2. We show that it is possible to define some general properties for a second order accurate scheme for numerical advection which ensure that the solution will not generate undesirable oscillations.

Having provided a satisfactory resolution of scalar advection, we then move on to examine linear hyperbolic systems of equations. This interest is motivated by the fact that many conservation laws that we are interested in are indeed hyperbolic systems. We carry out our study in two easy stages. In the first stage, we study linear hyperbolic systems, which give rise to a system of waves that are easy to analyze. We will show that the evolution of a linear hyperbolic system can be written in terms of a sequence of advection equations. The advected variable is indeed the characteristic variable and it is advected with a speed given by the eigenvalue. Concepts that we study in this chapter, like simple waves and the Riemann problem, will become increasingly important in the next few chapters when we study non-linear hyperbolic systems. It is for this reason that linear hyperbolic systems form a bridge that will lead us to our study of non-linear hyperbolic systems. Linear hyperbolic systems will, therefore, be the object of our study in the latter part of this Chapter. The second stage of our study, which we will undertake in the next few chapters, will consist of understanding how to deal with non-linear hyperbolic systems.

Our study of the simpler, linear hyperbolic systems will yield several important insights, many of which go over to our study of non-linear hyperbolic systems. Sub-Section 3.4.1 catalogues several properties of linear hyperbolic systems, making a clear distinction between those properties they share with their non-linear cousins and those that they do not. The Riemann problem for linear hyperbolic systems is also discussed in Sub-section 3.4.2. Sub-section 3.4.3 shows how the Riemann problem can be used as a building block for the numerical solution of hyperbolic systems. Section 3.5 tells us how the structure of the waves in a linear hyperbolic system dictate the form of the numerical boundary conditions. Sub-section 3.6.1 discusses solution techniques for linear hyperbolic systems using a modified Lax-Wendroff approach. Sub-section 3.6.2 discusses the implementation of a two-step Runge-Kutta scheme with characteristic limiting and Riemann solvers for linear hyperbolic systems. Sub-section 3.6.3 presents a predictor-corrector scheme with characteristic limiting and Riemann solvers for linear hyperbolic systems. Numerical results from these schemes are shown in Sub-section

3.6.4. Section 3.6 is useful because many of the items that we learn in that section constitute the building-blocks of schemes for solving non-linear hyperbolic systems.

3.2) Qualitative Introduction to Non-Linear Hybridization for Scalar Advection

In Section 2.7 we tried to construct several *linear* schemes for numerically evolving the scalar advection equation $u_t + a\,u_x = 0$, which is indeed the simplest prototypical linear hyperbolic system. We saw, however, that all of them proved dissatisfactory in one way or another. The linear donor cell scheme could advect square pulses without adding extra wiggles but it was very dissipative and dispersive. The linear second order schemes, i.e. the Lax-Wendroff and Runge-Kutta schemes, could advect smooth pulses with much reduced dissipation and dispersion but only at the expense of generating spurious oscillations at discontinuities. Such undershoots can be fatal if the variable being advected is an inherently positive quantity such as fluid density. Within the strict limits set by the Lax-Richtmeyer theorem, it seemed impossible to find linear schemes that could advect smooth profiles as well as discontinuities with high order accuracy. Historically speaking, this remained a stumbling block in scheme design for a long time. Godunov (1959) had already proved a theorem, known as *Godunov's theorem*, which effectively said that there are no *linear*, second order accurate schemes for treating *linear* advection which would always retain positivity of the solution. The full import of Godunov's theorem would remain unappreciated for almost twenty years. The distinction of finding a way out of this conundrum fell on Bram van Leer, an astronomer working at Leiden in the 1970s. van Leer (1977, 1979) realized that the way out of this dilemma lay in designing inherently *non-linear* schemes for treating the linear advection problem, thereby escaping the clutches of Godunov's theorem! This idea of using an inherently non-linear scheme is sometimes referred to as *non-linear hybridization*. van Leer went on to become a much-celebrated professor of aeronautical engineering and his seminal papers on advection and fluid flow have been cited several thousand times. Let us, therefore, retrace some of the thinking that led to successful schemes for linear advection.

Starting from an old idea of Godunov (1959), van Leer realized that one could think of the fluid in a computational zone as representing a physical slab of fluid. Let us focus on solution techniques for $u_t + a\, u_x = 0$, where we think of "u" as being some sort of fluid density that has to be advected to the right with a speed "a". In one dimension we may, if we wish, take "u" to have units of fluid mass per unit length. To simplify Figs 3.1 to 3.4 in this section, we will use a=1 and mesh size $\Delta x = 1$in those figures and we will also specialize our CFL number so that $\mu \equiv a\,\Delta t/\Delta x = \Delta t = 0.4$. The formulae that we derive will, however, be free of this restriction. Fig. 3.1 is based on Godunov's idea of moving slabs of fluid and shows the evolution of the fluid during a timestep that goes from t^n to $t^{n+1} = t^n + \Delta t$. The solid line in Fig. 3.1a shows five such slabs of fluid for five zones of a mesh at time t^n. The zone centers are given by $x_i = (i - 1/2)\,\Delta x$ and the zone boundaries are given by $x_{i+1/2} = i\,\Delta x$. In Fig. 3.1 the zone centers are labeled with the index "*i*". The zone-averaged values of the mesh function at time t^n are given by $\bar{u}^n_{i-2} = 1$, $\bar{u}^n_{i-1} = 1$, $\bar{u}^n_i = 0.25$, $\bar{u}^n_{i+1} = 0.1$ and $\bar{u}^n_{i+2} = 0.1$. We will focus specifically on the i^{th} zone and its immediate neighbors because they straddle a discontinuity. By interpreting advection with a>0 as being, quite literally, a rightward shift in the profile of the fluid by a distance of "$a\,\Delta t$" , the dashed line in Fig. 3.1b shows the original mesh function shifted by four-tenths of a mesh size (since we are using a timestep of size $\Delta t = 0.4$ and a CFL number of 0.4 in these figures). The mesh itself does not move. At the end of the timestep, i.e. at time t^{n+1} , we want the fluid to be represented as zone-averaged slabs of fluid on the mesh. We accomplish this by evaluating the total fluid under the dashed line in Fig. 3.1b within each zone and dividing it out by Δx to obtain a zone-averaged fluid density within each zone of interest. This is shown by the solid lines in Fig. 3.1b. A quantitative understanding of advection can be obtained by realizing that the light gray rectangle in Fig. 3.1b shows the amount of fluid that has crossed over into zone "*i*" from its left boundary. In other words, the light gray rectangle depicts a time-average of the left flux so that we have

$$\Delta t\ \bar{f}^{n+1/2}_{i-1/2} = (a\,\Delta t)\ \bar{u}^n_{i-1} \qquad (3.1)$$

where $\overline{\mathrm{f}}_{i-1/2}^{n+1/2}$ is the mean flux of fluid passing through zone boundary $x_{i-1/2}$. The dark gray rectangle in Fig. 3.1b shows the amount of fluid that has flowed out of zone "*i*" from its right boundary. In other words, the dark gray rectangle depicts a time-average of the right flux so that we have

$$\Delta t\ \overline{\mathrm{f}}_{i+1/2}^{n+1/2} = (\mathrm{a}\ \Delta t)\ \overline{\mathrm{u}}_i^n \tag{3.2}$$

where $\overline{\mathrm{f}}_{i+1/2}^{n+1/2}$ is the mean flux of fluid going through zone boundary $x_{i+1/2}$. We can then account for the fluid at time t^{n+1} in zone "*i*" by asserting a conservation law for the fluid in integral form as

$$\Delta x\ \overline{\mathrm{u}}_i^{n+1} = \Delta x\ \overline{\mathrm{u}}_i^n + \Delta t\ \overline{\mathrm{f}}_{i-1/2}^{n+1/2} - \Delta t\ \overline{\mathrm{f}}_{i+1/2}^{n+1/2} \quad \Leftrightarrow \quad \overline{\mathrm{u}}_i^{n+1} = \overline{\mathrm{u}}_i^n - \mu\left(\overline{\mathrm{u}}_i^n - \overline{\mathrm{u}}_{i-1}^n\right) = \left(1-\mu\right)\overline{\mathrm{u}}_i^n + \mu\overline{\mathrm{u}}_{i-1}^n \tag{3.3}$$

The reader is encouraged to review eqns. (2.2) and (2.3) and relate them to the above three equations.

We see that eqn. (3.3) retrieves the first order accurate upwind scheme, i.e. the donor cell scheme. Because, the donor scheme for linear advection is entirely equivalent to shifting around featureless slabs of fluid, we see that it does not generate any new oscillations at the end of the timestep that were not present at the beginning of the timestep. Viewed mathematically, each $\overline{\mathrm{u}}_i^{n+1}$ in eqn. (3.3) is a convex combination of its neighbors at the previous time. In other words, notice that in the equation $\overline{\mathrm{u}}_i^{n+1} = \left(1-\mu\right)\overline{\mathrm{u}}_i^n + \mu\overline{\mathrm{u}}_{i-1}^n$ from eqn. (3.3), the coefficients $\left(1-\mu\right)$ and μ are both positive for $0 < \mu < 1$. A scheme with such a property of not generating any new extrema in the solution that were not present initially is called a *monotonicity preserving scheme*. A more formal, yet conceptually equivalent, definition of a monotonicity preserving scheme is that if $\overline{\mathrm{u}}_i^n$ lies between $\overline{\mathrm{u}}_{i-1}^n$ and $\overline{\mathrm{u}}_{i+1}^n$ then the scheme ensures that $\overline{\mathrm{u}}_i^{n+1}$ lies between $\overline{\mathrm{u}}_{i-1}^{n+1}$ and $\overline{\mathrm{u}}_{i+1}^{n+1}$ for all zones "*i*". For linear advection with $0 < \mu < 1$ a sufficient condition

would consist saying that if $\bar{u}_i^n$ lies between $\bar{u}_{i-1}^n$ and $\bar{u}_{i+1}^n$ then the scheme should guarantee that $\bar{u}_i^{n+1}$ lies between $\bar{u}_{i-1}^n$ and $\bar{u}_i^n$ for all zones "*i*" . Similarly, for $-1 < \mu < 0$, a sufficient condition would require that if $\bar{u}_i^n$ lies between $\bar{u}_{i-1}^n$ and $\bar{u}_{i+1}^n$ then the scheme should guarantee that $\bar{u}_i^{n+1}$ lies between $\bar{u}_{i+1}^n$ and $\bar{u}_i^n$ for all zones "*i*". The positivity of the coefficients of $\bar{u}_i^n$ and $\bar{u}_{i-1}^n$ in eqn. (3.3) shows us that the donor cell scheme is also *positivity preserving*. Indeed, for scalar advection, the positivity preserving and monotonicity preserving properties are identical. This is because we can always either add a constant value to a mesh function to obtain another mesh function or flip the sign of a mesh function to obtain another mesh function. The resulting mesh functions also satisfy the scalar advection equation with the result that if the monotonicity preserving property is enforced then positivity is ensured and vice versa. The donor cell scheme in eqn. (3.3) is first order accurate and, therefore, very diffusive. For that reason, we wish to explore monotonicity preserving schemes which are second order accurate extensions of the donor cell scheme.

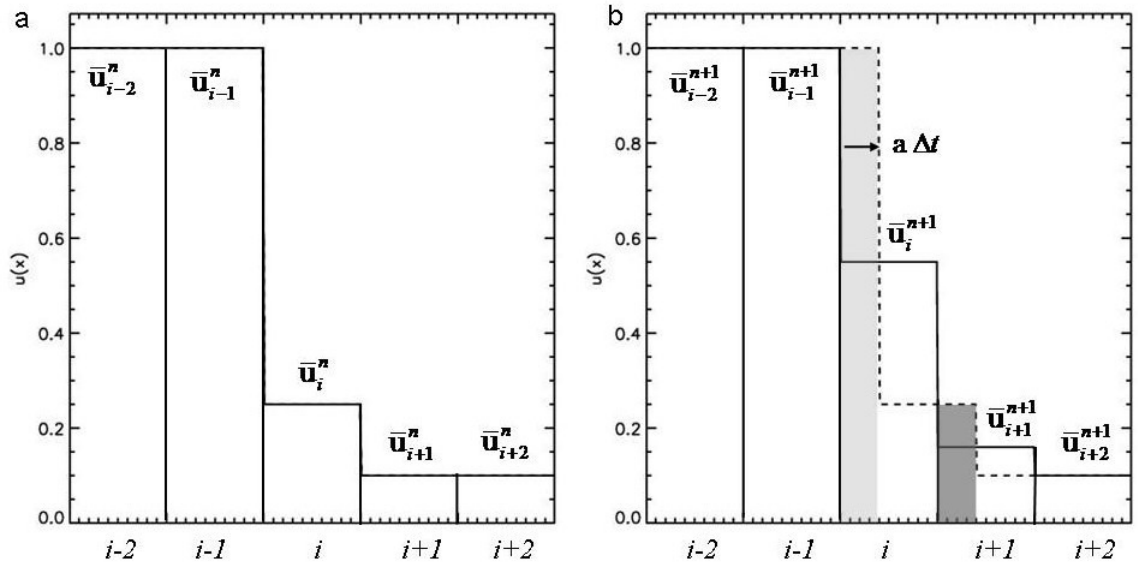


*Fig. 3.1 depicts a single timestep in the advection of fluid, where the fluid is modeled as a constant slab in each zone. We use Δx=1, Δt=0.4 and an advection speed "*a*" of unity. Fig. 3.1a shows the initial slabs of fluid with solid lines. Fig. 3.1b shows the slabs of fluid after they have been advected (dashed line) and the final profile of the mesh function at the end of the timestep (solid line). The total amount of fluid entering zone "i" from the left is shaded light gray. The total amount of fluid exiting zone "i" to the right is shaded dark gray.*

The previous paragraphs enable us to realize that the donor cell scheme has a low order of accuracy because the slabs are represented as featureless profiles. We realize though that we can look to the zones on either side of a zone of interest and, therefore, evaluate a slope within each slab. Different slopes can be built based on whether we take right-biased finite differences $\overline{\mathrm{u}}_{i+1}^{n} - \overline{\mathrm{u}}_{i}^{n}$, left-biased finite differences $\overline{\mathrm{u}}_{i}^{n} - \overline{\mathrm{u}}_{i-1}^{n}$ or central differences $\left(\overline{\mathrm{u}}_{i+1}^{n} - \overline{\mathrm{u}}_{i-1}^{n}\right)\big/2$. Fig. 3.2a shows us the same mesh function as Fig. 3.1a but this time we have looked to the right of each zone and built a right-biased slope within each zone "*i*" . Thus within each zone we evaluate an *undivided difference* $\overline{\Delta \mathrm{u}_i}^{n} = \overline{\mathrm{u}}_{i+1}^{n} - \overline{\mathrm{u}}_{i}^{n}$ so that the slope in zone "*i*" is given by $\overline{\Delta \mathrm{u}_i}^{n}\big/\Delta x$. The undivided differences can also be thought of as slopes that have been multiplied by Δx and are often referred to as just the *slopes*. This process of endowing the slabs of fluid on a mesh with internal structure is known as *reconstruction* and we say that we have carried out *piecewise linear reconstruction* within each zone. The piecewise linear reconstruction of the mesh function is shown as a dashed profile in Fig. 3.2a where the right biased finite differences have been used to evaluate the slopes. The fluid's profile in zone "*i*" is written as

$$\mathrm{u}_{i}^{n}(x) \;=\; \overline{\mathrm{u}}_{i}^{n} \;+\; \frac{\overline{\Delta \mathrm{u}_i}^{n}}{\Delta x}\,\left(x - x_i\right) \tag{3.4}$$

Advecting the fluid with second order accuracy is now tantamount to shifting the piecewise linear profile in Fig. 3.2a to the right by a distance "$\mathrm{a}\,\Delta t$" . The shifted profile is shown by the dashed line in Fig. 3.2b. The final step in the second order advection now consists of spatially averaging the shifted profile over each of the zones. This yields the solution at time t^{n+1} and is shown by the solid slabs in Fig. 3.2b. A quantitative understanding of piecewise linear advection can be obtained by realizing that the area of the light gray trapezoid in Fig. 3.2b shows the amount of fluid that has crossed over into zone "*i*" from its left boundary. The base of the trapezoid is $(\mathrm{a}\,\Delta t)$, the height of its left

side is $\overline{u}_{i-1}^{n} + (1/2 - \mu)\overline{\Delta u}_{i-1}^{n}$ and the height of its right side is $\overline{u}_{i-1}^{n} + \overline{\Delta u}_{i-1}^{n}/2$, making the area easy to evaluate. The time-average of the left flux, $\overline{f}_{i-1/2}^{n+1/2}$, is then given by:

$$\Delta t\ \overline{f}_{i-1/2}^{n+1/2} = (a\ \Delta t)\left[\overline{u}_{i-1}^{n} + \frac{1}{2}(1-\mu)\overline{\Delta u}_{i-1}^{n}\right] \tag{3.5}$$

where $\overline{f}_{i-1/2}^{n+1/2}$ is the mean flux of fluid going through zone boundary $x_{i-1/2}$. The dark gray trapezoid in Fig. 3.2b shows the amount of fluid that has flowed out of zone "*i*" from its right boundary. I.e. it is a time-average of the right flux so that we have

$$\Delta t\ \overline{f}_{i+1/2}^{n+1/2} = (a\ \Delta t)\left[\overline{u}_{i}^{n} + \frac{1}{2}(1-\mu)\overline{\Delta u}_{i}^{n}\right] \tag{3.6}$$

where $\overline{f}_{i+1/2}^{n+1/2}$ is the mean flux of fluid going through zone boundary $x_{i+1/2}$. We can then account for the fluid at time t^{n+1} in zone "*i*" by writing an integral expression for the conservation of fluid in that zone. Our general second order scheme is therefore given by

$$\begin{aligned} &\Delta x\ \overline{u}_{i}^{n+1} = \Delta x\ \overline{u}_{i}^{n} + \Delta t\ \overline{f}_{i-1/2}^{n+1/2} - \Delta t\ \overline{f}_{i+1/2}^{n+1/2} \quad \Leftrightarrow \\ &\overline{u}_{i}^{n+1} = \overline{u}_{i}^{n} - \mu\left(\overline{u}_{i}^{n} - \overline{u}_{i-1}^{n}\right) - \frac{\mu}{2}(1-\mu)\left(\overline{\Delta u}_{i}^{n} - \overline{\Delta u}_{i-1}^{n}\right) \end{aligned} \tag{3.7}$$

Fig. 3.2b clearly shows that the scheme in eqn. (3.7) has produced a spurious extremum in zone "i-1" at time t^{n+1}.

By comparing eqn. (3.7) to eqn. (3.3) we see that the use of piecewise linear profiles has produced an extra term in the update equation that depends on the slopes. It is also interesting to realize that our choice of right-biased slopes indeed yields the Lax-Wendroff scheme from Sub-section 2.7.3. We know that the Lax-Wendroff scheme is not monotonicity preserving from Fig. 2.19b, where we saw that it can produce large overshoots and undershoots that were not present in the original solution. By examining the solid lines in Fig. 3.2b we see that the advection has produced a spurious overshoot in

zone "i–1" at time t^{n+1} . It is easy to see that the piecewise linear profile that we introduced in zone "i" has not had a deleterious effect while the piecewise linear profile that we introduced in zone "*i-1*" has indeed spawned an overshoot in the same zone. We are inclined to ask what makes one linear profile acceptable when the other is not? The answer emerges when we realize that the linear profile that we introduced in zone "*i-1*" produces an overshoot, i.e. a local maximum, that is not present in the original mesh function. That is not the case for the profile that we introduced in zone "i". Once an unphysical extremum has formed in one timestep, its further growth will be exacerbated in subsequent timesteps, resulting in larger overshoots or undershoots. The utility of Fig. 3.2 stems from the fact that it enables us to get a visual understanding of the process by which an overshoot is formed in one timestep. We see that the overshoot comes about because the reconstructed piecewise linear profile itself has an overshoot – an overshoot that was not present in the original mesh function in Fig. 3.2a. Choosing a centered slope (or even a left-biased slope) would not solve the problem. Fig. 3.3 shows the consequence of using centered slopes, resulting in a scheme by Fromm (1968). We see that the problem persists. Using left-biased slopes yields the Beam-Warming scheme. Using the Beam-Warming scheme with the same initial conditions would have resulted in an undershoot in zone "i+1" at time t^{n+1} . We therefore see that all second order accurate schemes that impart an unrestricted, piecewise linear profile within the zones are doomed to produce overshoots or undershoots. The solution consists of restricting, i.e. *limiting*, the piecewise linear profile within each zone so that it does not produce any *new* extrema that were not initially present in the original slabs of fluid. We examine strategies for limiting the slopes next.

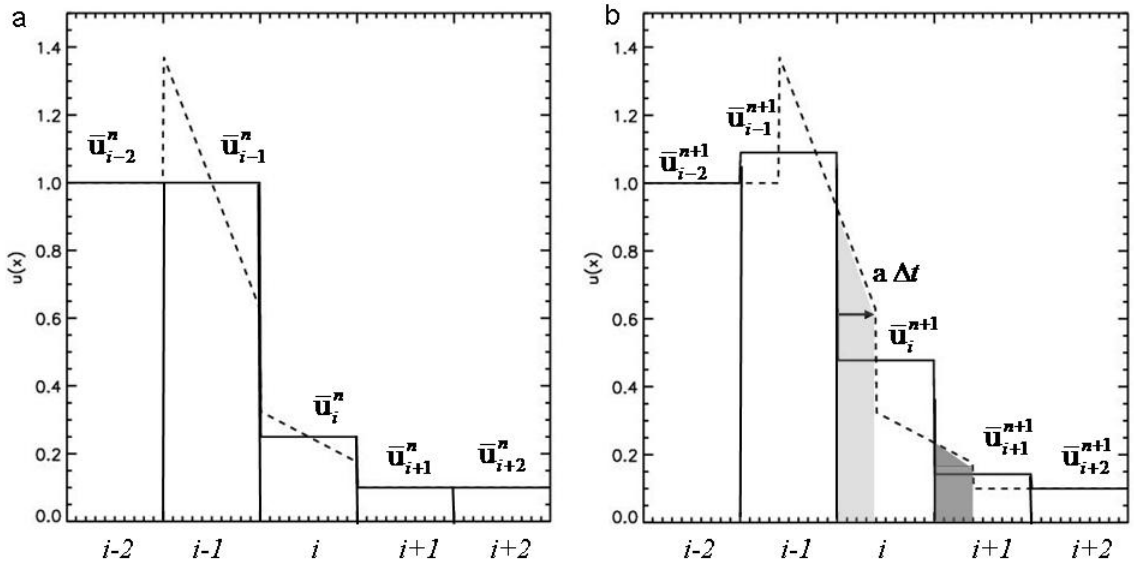


*Fig. 3.2 depicts a single timestep in the advection of fluid, where the fluid is modeled as a piecewise linear profile in each zone. The linear profiles were evaluated using right-biased slopes, making this a depiction of advection by the Lax-Wendroff scheme. We use $\Delta x=1$, $\Delta t=0.4$ and an advection speed "*a*" of unity. Fig. 3.2a shows the initial slabs of fluid with solid lines. The dashed line in Fig. 3.2a shows the piecewise linear reconstruction. Fig. 3.2b shows the slabs of fluid after they have been advected (dashed line) and the final profile of the mesh function at the end of the timestep (solid line). The total amount of fluid entering zone "i" from the left is shaded light gray. The total amount of fluid exiting zone "i" to the right is shaded dark gray.*

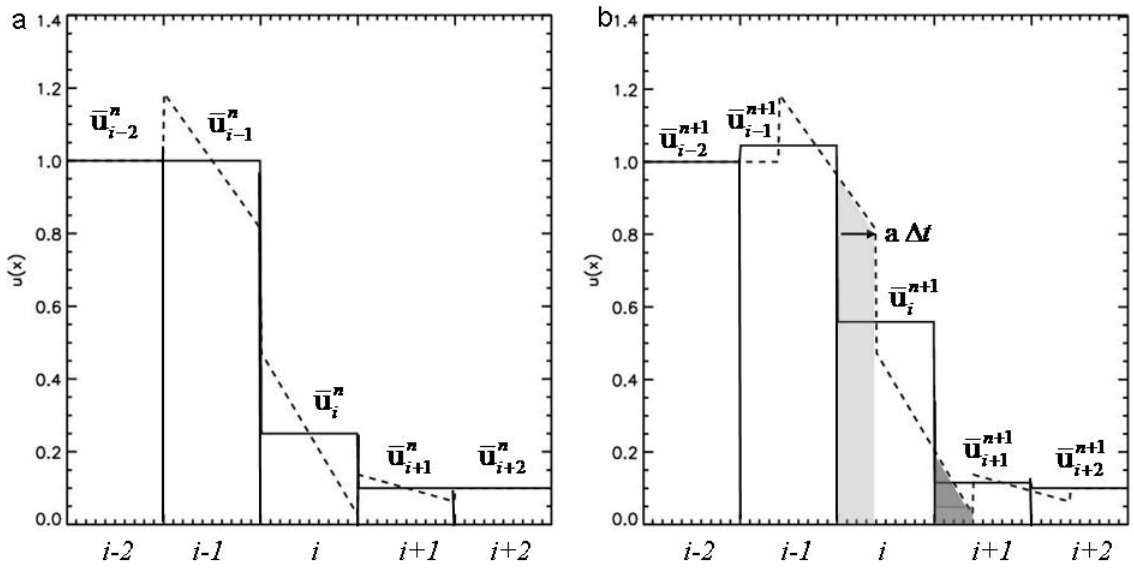


*Fig. 3.3 depicts a single timestep in the advection of fluid, where the fluid is modeled as a piecewise linear profile in each zone. The linear profiles were evaluated using centered slopes, making this a depiction of advection by the Fromm scheme. We use $\Delta x=1$, $\Delta t=0.4$ and an advection speed "*a*" of unity. Fig. 3.3a shows the initial slabs of fluid with solid lines. The dashed line in Fig. 3.3a shows the piecewise linear reconstruction. Fig. 3.3b shows the slabs of fluid after they have been advected (dashed line) and the final profile of the mesh function at the end of the timestep (solid line). The total amount of fluid entering zone "i" from the left is shaded light gray. The total amount of fluid exiting zone "i" to the right is shaded dark gray.*

Fig. 3.2a clearly shows that the spurious extremum in the linear reconstruction, i.e. the dashed line in Fig. 3.2a, could have been avoided if we had taken the smaller of the left and right-biased slopes in all the zones. The left-biased slope, also referred to as the left slope, in zone "*i*–1" would then be zero. So even though the right slope in zone "*i*–1" is rather large, we can use our knowledge of the left slope to limit the overall piecewise linear slope that we impart to the zone "*i-1*" in Fig. 3.2a. A similar line of reasoning can be used to limit the slopes in Fig. 3.3a. Fig. 3.4a shows the same initial profile but this time the undivided differences in all the zones have been evaluated in accordance with the algorithm

$$\overline{\Delta \mathrm{u}}_i^n = \frac{1}{2}\left(\mathrm{sgn}\left(\overline{\mathrm{u}}_{i+1}^n - \overline{\mathrm{u}}_i^n\right) + \mathrm{sgn}\left(\overline{\mathrm{u}}_i^n - \overline{\mathrm{u}}_{i-1}^n\right)\right) \min\left(\left|\overline{\mathrm{u}}_{i+1}^n - \overline{\mathrm{u}}_i^n\right|, \left|\overline{\mathrm{u}}_i^n - \overline{\mathrm{u}}_{i-1}^n\right|\right) \qquad (3.8)$$

Here the function sgn (x) is +1 for $x \geq 0$ and -1 for x<0. Notice that if the left and right undivided differences have the same sign then $\overline{\Delta u}_i^n$ will also have that sign. It will then have a magnitude that is given either by the absolute value of the right undivided difference, $\left|\overline{u}_{i+1}^n - \overline{u}_i^n\right|$, or the left undivided difference , $\left|\overline{u}_i^n - \overline{u}_{i-1}^n\right|$, depending on which one has the smaller value. Notice that the function depicted on the right hand side of eqn. (3.8) takes two arguments, i.e. the right- and left-biased undivided differences, and produces one value for the final undivided difference $\overline{\Delta u}_i^n$ for use in the numerical scheme in eqn. (3.7). It limits the undivided differences that are used in making the slope and is, therefore, referred to as a *slope limiter*. The specific function shown in eqn. (3.8) is called a *minmod slope limiter* and there are several similarly designed slope limiters. The slope limiters depend on the mesh-function in a nonlinear fashion. Consequently, schemes that use limiters for linear advection are inherently nonlinear. Since all slope limiters are extremely non-linear functions of their arguments, they help justify our claim that our successful schemes for linear advection are based on *non-linear hybridization.* The box at the end of this section catalogues several slope limiters. Interestingly, eqn. (3.8) is also the most computationally efficient implementation of the minmod limiter. We see from the dashed profile in Fig. 3.4a that the minmod limiter has eliminated all spurious overshoots and undershoots in the piecewise linear profile. Fig. 3.4b shows the result of advecting the dashed profile in Fig. 3.4a. We see that the final advected profile in Fig. 3.4b is free of overshoots and undershoots.

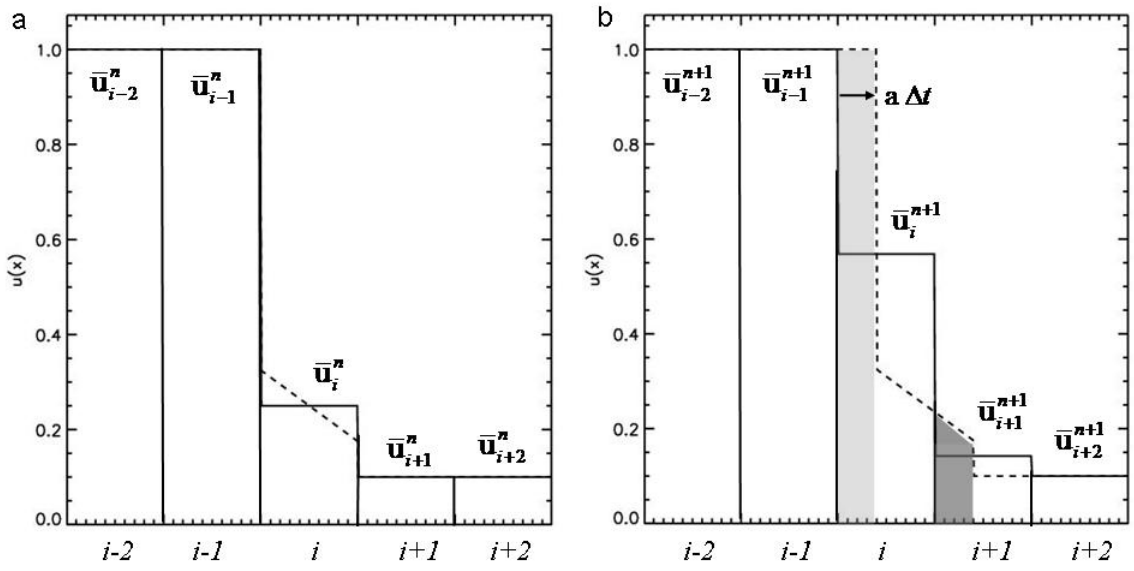


*Fig. 3.4 depicts a single timestep in the advection of fluid, where the fluid is modeled as a piecewise linear profile in each zone. The linear profiles were evaluated using the MinMod limiter. We use Δx=1, Δt=0.4 and an advection speed "*a*" of unity. Fig. 3.4a shows the initial slabs of fluid with solid lines. The dashed line in Fig. 3.4a shows the piecewise linear reconstruction. Fig. 3.4b shows the slabs of fluid after they have been advected (dashed line) and the final profile of the mesh function at the end of the timestep (solid line). The total amount of fluid entering zone "i" from the left is shaded light gray. The total amount of fluid exiting zone "i" to the right is shaded dark gray.*

Notice from the previous paragraph that the minmod limiter only achieved its success by locally clipping the solution. As a result, the order of the method is locally reduced by a limiter in those zones where it is activated. In other words, the limiter achieves its salutary effect by providing strong dissipation where it is needed to prevent dispersive ripples that would otherwise form in a second order scheme. (Recall from your basic physics studies that dispersion refers to the fact that waves with different wavelengths propagate at different speeds. The dashed profile with overshoots in Fig. 3.2a has a lot of short wavelength information in it owing to the overshoot. It is this short wavelength information that propagates on the mesh at speeds that are substantially different from the advection speed "a". This results in the formation of new extrema in the next timestep, as shown in Fig. 3.2b. The limiter locally reduces dispersion at the expense of locally enhancing dissipation.) Taking Fig. 3.4a as an example, we see that applying the limiter to zone "*i–1*" has made the flux at zone boundary "*i–1/2*" a first order accurate flux; the flux at zone boundary "*i+1/2*" remains second order accurate

because the limiter left the slope in zone "*i*" unchanged. Notice that the limiter achieves its beneficial goal by examining the solution over a larger stencil than one would have in a straightforward, second order accurate Lax-Wendroff scheme. The limiter in eqn. (3.8) effectively compares the ratio of the left undivided difference, given by $\overline{u}_i^n - \overline{u}_{i-1}^n$, to the right undivided difference, given by $\overline{u}_{i+1}^n - \overline{u}_i^n$, in its effort to pick out an undivided difference with the smaller absolute value. Consequently, the important quantity when examining whether the solution has a local extremum is given by taking the ratio of the left slope to the right slope as follows

$$\theta_i = \frac{\overline{u}_i^n - \overline{u}_{i-1}^n}{\overline{u}_{i+1}^n - \overline{u}_i^n} \tag{3.9}$$

Negative values of θ_i cause the resulting slope to be zero. When θ_i is close to unity, the mesh function is taken to be reasonably smooth and the undivided difference provided by a good slope limiter should revert back to the centered finite difference. As θ_i becomes much larger or much smaller than unity the mesh function is taken to have a significant kink in it. When θ_i begins deviating strongly from unity, an increasing amount of slope limiting is provided by the limiter. Notice that the minmod limiter clips the slope in all situations where θ_i makes even the slightest deviation from unity. Some of the other limiters, which are catalogued in the box at the end of this section, permit a little more latitude.

We should never forget though that the solution may have a local extremum that could be physically meaningful and the minmod limiter, like all other limiters, will clip that local extremum. Osher and Chakravarthy (1984) have indeed shown that any method that uses the limiters described here must degenerate to first order of accuracy at points of local extrema. Their result makes sense because a simple ratio of slopes, as in eqn. (3.9), cannot give us a good diagnostic of the structure of the flow. For that reason, a lot of research has gone into designing schemes that are gentler in the manner in which they reduce undesired oscillations. A class of such schemes that succeeds in this regard is

known as the Weighted Essentially Non-Oscillatory schemes (WENO henceforth), but we will defer the study of WENO schemes to Chapter 7. The Piecewise Parabolic Method (PPM henceforth) also pays a lot of careful attention to the reconstruction problem and is also presented in Chapter 7. Even amongst the monotonicity preserving limiters, there are several choices and some of the better ones produce solutions whose quality is discernably better than the poorer choices. In fact, the humble minmod limiter from eqn. (3.8) is one of the most dissipative limiters. Yet it is not to be scoffed at, because when the physical problem becomes very stringent the minmod limiter can indeed become the most trusted choice.

We now apply the minmod limiter and one of its more sophisticated cousins to the advection problems that we first presented in Section 2.7. The more sophisticated limiter that we use goes under the name of the *monotonized central difference limiter* (MC henceforth) and was also proposed by van Leer (1977). The MC limiter is catalogued in the box at the end of this section. As before, we set the propagation speed to unity and propagate certain profiles around the unit interval, $x \in [-0.5, 0.5]$ with periodic boundary conditions. We used a CFL number of 0.8. Our first profile consists of a Gaussian profile $\mathrm{u}(x, t=0) = e^{-(x/0.1)^2}$ which is plotted in Fig. 3.5a at times t=0 (solid line), t=1 (asterisks) and t=2 (diamonds). Our second profile consists of setting $\mathrm{u}(x, t=0) = 1 \; \forall \; x \in [-0.05, 0.05]$ and $\mathrm{u}(x, t=0) = 0$ elsewhere. It is shown in Fig. 3.5b at times of t=0 (solid line), t=0.25 (asterisks) and t=0.75 (diamonds). The Gaussian, because it is smooth, was advected on a 50 zone mesh whereas the square pulse was advected on a 100 zone mesh. Fig. 3.5 is based on the advection scheme from eqn. (3.7) with the minmod limiter applied to the slopes. Fig. 3.6 parallels Fig. 3.5 but uses the MC limiter. We see that Figs. 3.5a and 3.6a are dramatic improvements over Figs. 2.17a and 2.21a for the first order accurate schemes in Section 2.7. Furthermore, they even eliminate the slight undershoot that plagued the Lax-Wendroff and Runge-Kutta schemes in Figs. 2.19a and 2.20a. On comparing Fig. 3.6a with Fig. 2.19a for the Lax-Wendroff scheme we see, however, that the top of the Gaussian is clipped. This result is in accordance with our expectation that the accuracy degrades to first order at local extrema.

The minmod limiter has not been able to distinguish between the local maximum in the Gaussian, which is indeed physical, and the spurious oscillations that could arise in the vicinity of a discontinuous solution. Consequently, the minmod limiter, just like all the other limiters presented in this section, indiscriminately eliminates both types of extrema. Because of its relatively unsophisticated design, the minmod limiter clips the top of the Gaussian to a noticeably greater extent than the MC limiter, as can be seen by comparing Fig. 3.5a to Fig. 3.6a.

We now turn our attention to Figs. 3.5b and 3.6b for the advection of the top hat profile. Observe that we have made a considerable improvement over all the results in Section 2.7 for this problem. We see that both limiters produce oscillation-free propagation of the top hat profile, however, the solution from the minmod limiter is much more diffusive. It can in fact be shown that the minmod limiter is the most diffusive of the second order accurate limiters. Use of the MC limiter yields a much better looking profile for the propagating square wave. Note though that the solution in Fig. 3.6b does exhibit some diffusion. By observing the solution in Fig. 3.6b we see that the front and rear parts of the square wave have become unsymmetrical. This is inevitable considering that our scheme is still an *upwind biased scheme*. In principle, the forward and backward facing parts of profiles in Fig. 3.6b should have remained symmetrical. In reality, the loss of symmetry is a direct consequence of the directional bias that we have built into the scheme. Modern schemes do however go a long way towards minimizing this upwind bias.

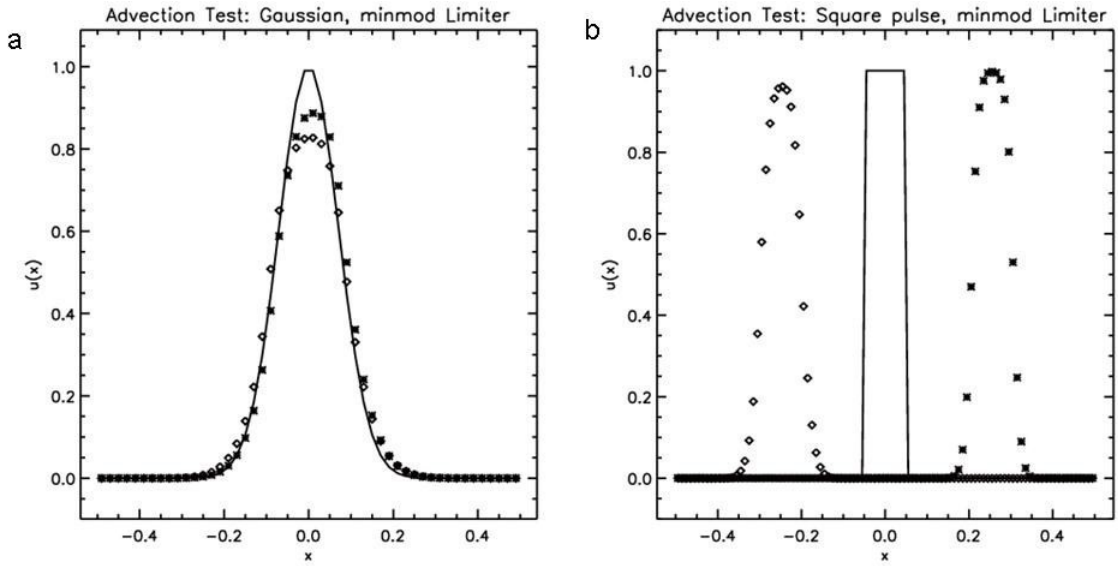


Fig 3.5a and 3.5b show the solutions from the second order scheme from eqn. (3.7) with a minmod limiter applied to the slopes. We simulate scalar advection with initial conditions given by a Gaussian profile and a top-hat profile respectively. The times in Fig. 3.5a are t=0 (solid line), t=1 (asterisks) and t=2 (diamonds). The times in Fig. 3.5b are t=0 (solid line), t=0.25 (asterisks) and t=0.75 (diamonds).

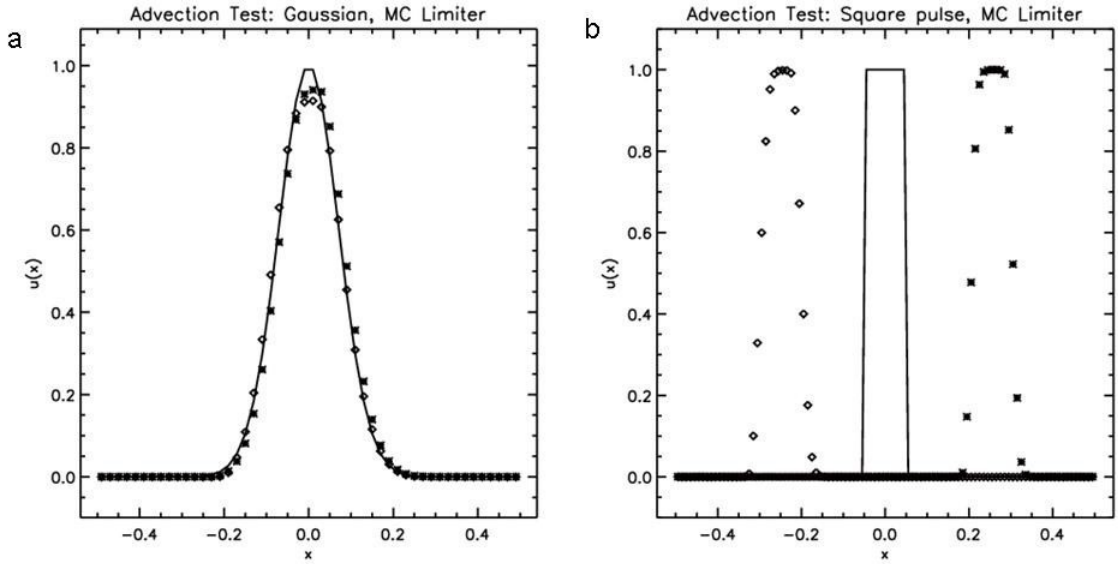


Fig 3.6a and 3.6b show the solutions from the second order scheme from eqn. (3.7) with an MC limiter applied to the slopes. We simulate scalar advection with initial conditions given by a Gaussian profile and a top-hat profile respectively. The times in Fig. 3.6a are t=0 (solid line), t=1 (asterisks) and t=2 (diamonds). The times in Fig. 3.6b are t=0 (solid line), t=0.25 (asterisks) and t=0.75 (diamonds).

More on Limiters

Observe from Figs. 3.2, 3.3 and 3.4 and eqns. (3.5) and (3.6) that changing the way the slope is constructed also changes the flux of fluid that is advected across zone boundaries. It is, therefore, possible to formulate the limiters in such a fashion as to make the limiting of the fluxes more evident. Such limiters are known as *flux limiters*, to contrast them with the slope limiters that we have used to illustrate the issues in this section. Since we will be emphasizing finite volume reconstruction as a building block for successful schemes in this book we will not use flux limiters in any of the practical schemes that we discuss later. It is, nevertheless, necessary to mention for the sake of completeness that for every slope limiter that one might formulate, there is a corresponding flux limiter that can operate directly on the fluxes.

Earlier in this section we had mentioned that the minmod is but one of a class of slope limiters. It helps to catalogue many of the popularly used limiters here along with their attribution. Thus with *a* and *b* specifying the left and right slopes respectively the slope limiters can be written as

Minmod (Roe 1986):

$$\text{minmod}\ (a,b) = \frac{1}{2}\ \left(\text{sgn}(a) + \text{sgn}(b)\right)\ \min\ \left(|a|, |b|\right)$$

van Leer (van Leer 1974):

$$\text{vanleer}\ (a,b) = \left(\text{sgn}(a) + \text{sgn}(b)\right)\ \frac{a\ b}{|a| + |b|}$$

Monotonized Central (MC)(van Leer 1977):

$$\text{MC}\ (a,b) = \frac{1}{2}\ \left(\text{sgn}(a) + \text{sgn}(b)\right)\ \min\ \left(\ \frac{1}{2}|a+b|,\ 2|a|,\ 2|b|\right)$$

MC_β:

$$\text{MC}_\beta\ (a,b) = \frac{1}{2}\ \left(\text{sgn}(a) + \text{sgn}(b)\right)\ \min\ \left(\ \frac{1}{2}|a+b|,\ \beta|a|,\ \beta|b|\right) \qquad 1 \le \beta \le 2$$

Superbee (Roe 1986):

$$\text{Superbee}\ (a,b) = \frac{1}{2}\ \left(\text{sgn}(a) + \text{sgn}(b)\right)\ \max\ \left(\ \min\ \left(2|a|, |b|\right)\ ,\ \min\ \left(|a|, 2|b|\right)\ \right)$$

Sweby (Sweby 1984):

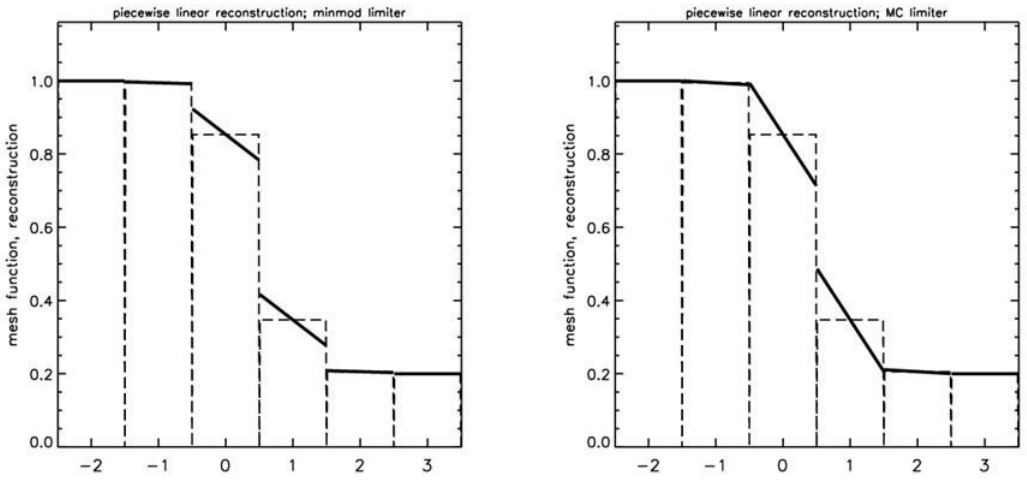
$$\text{Superbee}_{\beta}\ (a,b) = \frac{1}{2}\left(\text{sgn}(a)+\text{sgn}(b)\right)\ \max\left(\min\ \left(\beta|a|,|b|\right)\ ,\ \min\ \left(|a|,\beta|b|\right)\ \right)\quad 1\le\beta\le 2$$

The limiters are given here in a form that is most efficient when implementing them on modern computers with modern languages. Notice that the MC class of limiters have the advantage that they can retrieve the centered slope $(a+b)/2$ when the left and right slopes do not constrain the slope limiting process. The centered slope is the most stable slope that one can provide for smooth variations in the flow. Compared to the left and right slopes, it is also the most accurate slope. The MC class of limiters provide a special advantage over the other limiters in the vicinity of smooth flow because they permit us to retrieve a centered slope. The minmod limiter is the most stable of these limiters in the presence of strong discontinuities, with the *vanLeer* and MC limiters also performing ably on large classes of problems. While the *superbee* limiter by Roe (1986) can produce charming results for certain types of linear advection problems, it can also be a temperamental performer on problems with strong shocks. The Sweby limiter tones it down.

Notice that the MC_{β} limiter reduces to the minmod limiter when $\beta=1$ and reverts to the MC limiter when $\beta=2$. The Sweby limiter has a similar attribute where it reduces to the minmod limiter with $\beta=1$ and reverts to the Superbee limiter when $\beta=2$. One may, therefore, ask what is being controlled by the parameter β. The slopes produced by these adjustable limiters become larger as β approaches 2. Let us, therefore, take another look at Figs. 3.5b and 3.6b. The reason the square wave profile looks crisper in Fig. 3.6b is that the MC limiter produces larger slopes. The larger slopes tend to preserve the form of the square wave, i.e. they compress the profile so as to make it look crisper. Consequently, all the limiters, except for the minmod limiter, are also called *compressive limiters* because they try to produce a somewhat larger slope in order to preserve discontinuous profiles more crisply.

The ensuing two figures show us the difference between the minmod and MC limiters graphically. The dashed line in both figures shows the mesh function. The solid

lines show the reconstructed profiles for the minmod and MC limiters in the figures to the left and right respectively. In this example, the slope produced by the MC limiter is twice as large as the slope produced by the minmod limiter. Without introducing any new extrema, the MC limiter has produced the steeper, i.e. more compressed, profile with smaller jumps at zone boundaries. Now recall that the donor cell scheme produces the largest jumps in the reconstructed profile at zone boundaries, which also makes it a very diffusive scheme. Thus the extent of the jump in the reconstructed profiles at zone boundaries correlates with the amount of diffusion in the scheme. Since smaller jumps at the zone boundaries result in decreased numerical diffusion, we see that the MC limiter is less diffusive than the minmod limiter. For the reconstructed profiles shown below, the minmod limiter introduces jumps, and therefore dissipative fluxes, at $x = -1/2,\ 1/2$ and $3/2$. The fluxes produced by using the MC limiter also introduce dissipation at $x = 1/2$, thus contributing to stability. But the MC limiter does not introduce extra dissipation at the other two zone boundaries. Consequently, the MC limiter produces sharper profiles.



The dashed lines show the mesh function and the solid lines show the piecewise linear reconstructed profile. The figure to the left was produced with the minmod limiter, the figure to the right was produced with the MC limiter.

There are several flow features, such as an entropy wave in hydrodynamics or an Alfven wave in ideal MHD, where the flow feature should ideally propagate unchanged over very long distances. By using the eigenvectors that were introduced in Chapter 1 it is possible to detect where such features occur in the flow. Compressive limiters can be very useful in designing schemes that allow such features to propagate over long distances on a computational mesh without much change. Schemes that pay more

attention to the reconstruction problem, like the PPM or WENO schemes which we will study later, offer an even more elegant solution to the problem of accurate advection.

3.3) The Total Variation Diminishing Property and Understanding the Limiters

In the previous section we saw that the monotonicity preserving property is essential for obtaining schemes for scalar advection that don't generate any more extrema at the end of a timestep than were present at the beginning of a timestep. It is hard to formulate the monotonicity preserving property in a mathematically cogent way. For that reason Harten (1983) invented the concept of *total variation diminishing* (TVD henceforth) schemes where the TVD property is easily formulated. Harten was able to show that if a scheme is TVD then it should also be monotonicity preserving. Let us therefore specify a mesh with zone size Δx and a zone-centered collocation. Let the index "i" label the zones . We solve the scalar advection equation $u_t + a\, u_x = 0$ on this mesh. To simplify the treatment of the fluxes at the boundaries of our domain, we treat the domain as infinite with a solution that tends to zero as $x \to \pm\infty$. We define the total variation of a mesh function $\bar{u}^n \equiv \left\{ ..., \bar{u}_{i-1}^n, \bar{u}_i^n, \bar{u}_{i+1}^n, ... \right\}$ at time t^n as

$$\mathrm{TV}\left(\bar{u}^n\right) = \sum_{i=-\infty}^{\infty} \left| \bar{u}_{i+1}^n - \bar{u}_i^n \right| \tag{3.10}$$

After application of the scheme in eqn. (3.7) we get the mesh function $\bar{u}^{n+1} \equiv \left\{ ..., \bar{u}_{i-1}^{n+1}, \bar{u}_i^{n+1}, \bar{u}_{i+1}^{n+1}, ... \right\}$ at time $t^{n+1} = t^n + \Delta t$ and we can define $\mathrm{TV}\left(\bar{u}^{n+1}\right)$ similarly to eqn. (3.10). The TVD property then says that

$$\mathrm{TV}\left(\bar{u}^{n+1}\right) \le \mathrm{TV}\left(\bar{u}^n\right) \tag{3.11}$$

One has only to imagine a square pulse at one time step and envision it developing wiggles at the next time step. Fig. 2.19b provides such an example. Each of the upswings and downswings in the wiggles contributes positively to the total variation. The total

variation for the solid curve in Fig. 2.19b is exactly 2, i.e. an upswing of unity followed by a downswing of unity. We can see that the total variation in the dashed curve is much greater than 2, showing that the total variation increased after several time steps in Fig. 2.19b. A scheme that satisfies eqn. (3.11) is called a *TVD scheme*. A TVD scheme would not produce such an increase in the variation, thus preventing the growth of spurious oscillations.

Harten (1983) proved an incredibly important theorem for TVD schemes, which we explain below. Harten observed that the general structure of the update equation in eqn. (3.7) can be written as

$$\bar{u}_i^{n+1} = \bar{u}_i^n - C_{-,i-1/2}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) + C_{+,i+1/2}\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right) \tag{3.12}$$

where $C_{-,i-1/2}$ and $C_{+,i+1/2}$ can be any functions that depend on the mesh function in a linear or non-linear fashion. For the Lax-Wendroff scheme, which is a linear scheme for advection, eqn. (3.7) gives us $C_{-,i-1/2} = \mu(1+\mu)/2$ and $C_{+,i+1/2} = -\mu(1-\mu)/2$ so that the coefficients only depend on the CFL number. For any advection scheme that achieves its stability via non-linear hybridization, we in fact expect $C_{-,i-1/2}$ and $C_{+,i+1/2}$ to have a non-linear dependence on the mesh function. By permitting $C_{-,i-1/2}$ and $C_{+,i+1/2}$ to be data-dependent, *Harten's theorem* is general enough to encompass non-linear schemes for solving the linear advection problem. Harten was able to assert his main theorem which says that : *When* $C_{-,i+1/2} \geq 0$, $C_{+,i+1/2} \geq 0$ *and* $C_{-,i+1/2} + C_{+,i+1/2} \leq 1$ *for all zones "i" in the update equation given by eqn. (3.12), the scheme is TVD*. A scheme that produces new maxima or minima when the solution is evolved in time would violate the TVD property. If we take the Lax-Wendroff scheme with $0 < \mu < 1$ as an example, we see that $C_{+,i+1/2} < 0$ thus proving that it is not a TVD scheme. If the solution is positive to begin with, a TVD scheme would prevent it from going negative at a later time. We thus see that the TVD property will also help a scheme to be positivity preserving.

The proof of Harten's theorem goes as follows. By resetting $i \to i+1$ in eqn. (3.12) and subtracting eqn. (3.12) from the resulting equation we obtain

$$\bar{u}_{i+1}^{n+1} - \bar{u}_i^{n+1} = C_{-,i-1/2}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) + \left(1 - C_{-,i+1/2} - C_{+,i+1/2}\right)\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right) + C_{+,i+3/2}\left(\bar{u}_{i+2}^n - \bar{u}_{i+1}^n\right) \tag{3.13}$$

Now if the coefficients of $\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right)$, $\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)$ and $\left(\bar{u}_{i+2}^n - \bar{u}_{i+1}^n\right)$ are non-negative then an interesting result prevails. By repeatedly applying the Schwartz inequality (i.e. $|a+b| \le |a| + |b|$) to eqn. (3.13) and summing over all indices "i", let us evaluate the total variation in the mesh function at time $t^{n+1} = t^n + \Delta t$. We see that

$$\begin{aligned} \mathrm{TV}\left(\bar{u}^{n+1}\right) \equiv \sum_{i=-\infty}^{\infty} \left|\bar{u}_{i+1}^{n+1} - \bar{u}_i^{n+1}\right| \le \sum_{i=-\infty}^{\infty} \left(1 - C_{-,i+1/2} - C_{+,i+1/2}\right)\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right| \\ + \sum_{i=-\infty}^{\infty} C_{-,i-1/2}\left|\bar{u}_i^n - \bar{u}_{i-1}^n\right| + \sum_{i=-\infty}^{\infty} C_{+,i+3/2}\left|\bar{u}_{i+2}^n - \bar{u}_{i+1}^n\right| \end{aligned} \tag{3.14}$$

Making unit shifts in the summation indices "*i*" for the last two sums in eqn. (3.14) gives us

$$\begin{aligned} \mathrm{TV}\left(\bar{u}^{n+1}\right) \equiv \sum_{i=-\infty}^{\infty} \left|\bar{u}_{i+1}^{n+1} - \bar{u}_i^{n+1}\right| \le \sum_{i=-\infty}^{\infty} \left(1 - C_{-,i+1/2} - C_{+,i+1/2}\right)\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right| \\ + \sum_{i=-\infty}^{\infty} C_{-,i+1/2}\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right| + \sum_{i=-\infty}^{\infty} C_{+,i+1/2}\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right| \\ = \sum_{i=-\infty}^{\infty} \left|\bar{u}_{i+1}^n - \bar{u}_i^n\right| \equiv \mathrm{TV}\left(\bar{u}^n\right) \end{aligned} \tag{3.15}$$

This completes our proof of Harten's theorem.

Harten's theorem is important for TVD schemes because several advection schemes, including the monotonicity preserving scheme that was sketched out in the previous section, can be cast (or recast) to have a structure that formally looks like eqn. (3.12). As a result, TVD schemes are an important building block in many successful

strategies for treating hyperbolic systems. Notice though that in making the update equation for a scheme resemble eqn. (3.12) the coefficients $C_{-,i-1/2}$ and $C_{+,i+1/2}$ can acquire a strongly nonlinear dependence on the mesh function. The further utility of Harten's theorem, therefore, lies in the fact that it will be used to gain deep insights into the workings of the limiter and the non-linear hybridization that it introduces into the numerical scheme.

Now let us put the schemes that we designed in the previous section in context with the help of Harten's theorem. In light of the update equation for scalar advection, i.e. the first equation in eqn. (3.7), we realize that a scheme will have the TVD property if its fluxes have a special form. For that reason, let us compare the Lax-Wendroff fluxes in eqns. (3.5) and (3.6) to the donor cell fluxes in eqns. (3.1) and (3.2). Notice that the fluxes defined by eqns. (3.1) and (3.2) are diffusive but the first order scheme that utilizes those fluxes will indeed satisfy the TVD property. Consequently, the extra terms in the fluxes in eqns. (3.5) and (3.6) are referred to as the *anti-diffusive fluxes*. For example, using the right-biased undivided differences, $\overline{\Delta \mathrm{u}}_{i-1}^{n} = \overline{\mathrm{u}}_{i}^{n} - \overline{\mathrm{u}}_{i-1}^{n}$, in eqn. (3.5), the anti-diffusive flux has the form: $\mathrm{a}(1-\mu)\left(\overline{\mathrm{u}}_{i}^{n} - \overline{\mathrm{u}}_{i-1}^{n}\right)/2$. Recall too that the right-biased differences give us the Lax-Wendroff scheme. For $0 \le \mu \le 1$ we see that the coefficient in front of $\left(\overline{\mathrm{u}}_{i}^{n} - \overline{\mathrm{u}}_{i-1}^{n}\right)$ has a sign that is opposite to that of a diffusion term. In other words, the coefficient "$\mathrm{a}(1-\mu)/2$" is positive which is opposite to the sign that a diffusion term would have for an update equation of the form $\overline{\mathrm{u}}_{i}^{n+1} = \overline{\mathrm{u}}_{i}^{n} - \Delta t \left(\overline{\mathrm{f}}_{i+1/2}^{n+1/2} - \overline{\mathrm{f}}_{i-1/2}^{n+1/2}\right)/\Delta x$. Hence the flux is called anti-diffusive and it performs the task of steepening the mesh function. If the anti-diffusive fluxes are allowed to have a linear dependence on their neighboring mesh function, the previous section has shown us that the scheme will not be monotonicity preserving. In order to have a TVD advection scheme, we see that the slopes in the second order accurate fluxes from eqns. (3.5) and (3.6) have to be limited. The next advance came from Sweby (1984) who studied how the anti-diffusive fluxes can be limited in a fashion that is consistent with Harten's theorem. See also Tadmor

(1988). He showed that such a study yields a systematic strategy for understanding all the limiters. In the next paragraph we retrace the most important aspects of Sweby's paper.

To begin with, realize that if the full extent of the undivided differences $\overline{\Delta u}_{i-1}^{n}$ and $\overline{\Delta u}_{i}^{n}$ from the second order accurate fluxes $\bar{f}_{i-1/2}^{n+1/2}$ and $\bar{f}_{i+1/2}^{n+1/2}$ of eqns. (3.5) and (3.6) are used then the resulting second order scheme will not be monotonicity preserving. To design a TVD scheme we have to limit the slopes $\overline{\Delta u}_{i-1}^{n}$ and $\overline{\Delta u}_{i}^{n}$ in eqns. (3.5) and (3.6) respectively. We therefore use a flux limiter $\phi(\theta_i)$ which depends on θ_i from eqn. (3.9) for each zone "*i*". In eqn. (3.5) we replace $\overline{\Delta u}_{i-1}^{n} \rightarrow \phi(\theta_{i-1})\ \left(\bar{u}_{i}^{n} - \bar{u}_{i-1}^{n}\right)$, i.e. we are using a limited form of the right slope. We similarly replace $\overline{\Delta u}_{i}^{n} \rightarrow \phi(\theta_i)\ \left(\bar{u}_{i+1}^{n} - \bar{u}_{i}^{n}\right)$ in eqn. (3.6). We then obtain the limited fluxes

$$\bar{f}_{i-1/2}^{n+1/2} = a\left[\bar{u}_{i-1}^{n} + \frac{1}{2}(1-\mu)\ \phi(\theta_{i-1})\ \left(\bar{u}_{i}^{n} - \bar{u}_{i-1}^{n}\right)\right] \tag{3.16}$$

and

$$\bar{f}_{i+1/2}^{n+1/2} = a\left[\bar{u}_{i}^{n} + \frac{1}{2}(1-\mu)\ \phi(\theta_{i})\ \left(\bar{u}_{i+1}^{n} - \bar{u}_{i}^{n}\right)\right] \tag{3.17}$$

Intuitively, we seek a limiter such that $\phi(\theta) = 0$ for $\theta < 0$ so that the creation of new extrema is avoided. (Within the confines of our TVD formulation it is not possible to distinguish between smooth extrema in the actual solution and spurious extrema resulting from numerical oscillations, so we play it safe and avoid all extrema.) We also wish to have $\phi(\theta) > 0$ for $\theta > 0$ so that the slope within each zone has the right orientation. In order to achieve second order accuracy in the limit where the solution is smooth and monotone, we also wish to have $\phi(\theta) \rightarrow 1$ as $\theta \rightarrow 1$. I.e. when the solution is smooth, monotone and free of kinks, our limited scheme reverts back to the second order accurate

Lax-Wendroff scheme. We know from Section 2.7 that the Lax-Wendroff scheme performs beautifully in such situations. The second order accurate scheme from eqn. (3.7) is, therefore, modified to become

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \mu\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) - \frac{\mu}{2}(1-\mu)\left[\phi(\theta_i)\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right) - \phi(\theta_{i-1})\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right)\right] \qquad (3.18)$$

Observe that with $\phi(\theta) = 0$ eqn. (3.18) reverts to a donor cell scheme, whereas with $\phi(\theta) = 1$ eqn. (3.18) it becomes the Lax-Wendroff scheme with right-biased undivided differences. With the functional dependence $\phi(\theta) = \theta$ it can also become the Beam-Warning scheme with left-biased undivided differences. So the inclusion of the limiters has endowed our scheme with a great deal of flexibility. Please also note that the inclusion of limiters $\phi(\theta_i)$ and $\phi(\theta_{i-1})$ that depend on the ratios θ_i and θ_{i-1} in the above equation also results in an expansion of the stencil for our numerical scheme. We still have to recast eqn. (3.18) in a form that conforms with eqn. (3.12) so that Harten's theorem may be applied to it. A first attempt at recasting eqn. (3.18) would be to write it as

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \left[\mu - \frac{\mu}{2}(1-\mu)\phi(\theta_{i-1})\right]\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) + \left[-\frac{\mu}{2}(1-\mu)\phi(\theta_i)\right]\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right) \qquad (3.19)$$

We see that the coefficient of $\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)$ will be negative for normal situations when $0 \le \mu \le 1$ and $\phi(\theta_i) \sim 1$. As a result, the form of eqn. (3.19) is unsuitable for Harten's theorem. With the help of eqn. (3.9), we now substitute $\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right) = \left(\bar{u}_i^n - \bar{u}_{i-1}^n\right)\big/\theta_i$ in eqn. (3.19) to obtain

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \left\{\mu + \frac{\mu}{2}(1-\mu)\left[\frac{\phi(\theta_i)}{\theta_i} - \phi(\theta_{i-1})\right]\right\}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) \qquad (3.20)$$

By relating eqn. (3.20) to eqn. (3.12) we get

$$C_{-,i-1/2} = \mu + \frac{\mu}{2}(1-\mu)\left[\frac{\phi(\theta_i)}{\theta_i} - \phi(\theta_{i-1})\right] \text{ and } C_{+,i+1/2} = 0 \tag{3.21}$$

We, therefore, see that an application of Harten's theorem requires $0 \le C_{-,i-1/2}$ and $C_{-,i-1/2} \le 1$. For $0 \le \mu \le 1$ the former condition, i.e. $0 \le C_{-,i-1/2}$, requires that the square bracket in eqn. (3.21) is greater than or equal to –2 . The latter condition, i.e. $C_{-,i-1/2} \le 1$, requires that the same square bracket be less than or equal to 2. Consequently, we see that insisting on the TVD property is tantamount to requiring that

$$-2 \le \frac{\phi(\theta_i)}{\theta_i} - \phi(\theta_{i-1}) \le 2 \tag{3.22}$$

for all possible values of θ_i and θ_{i-1} . Problem 3.5 at the end of this chapter gently steps the reader through the derivation of eqn. (3.22). In general, θ_i and θ_{i-1} are independent of each other. Thus the only way to guarantee the TVD property is to have a non-negative function $\phi(\theta)$ which is bounded by

$$0 \le \frac{\phi(\theta)}{\theta} \le 2 \quad \text{and} \quad 0 \le \phi(\theta) \le 2 \tag{3.23}$$

Eqn. (3.23) identifies the region where the TVD property holds. For positive flux limiters, this region is bounded by $\phi(\theta) \le 2\theta$ and $\phi(\theta) \le 2$. Fig. 3.7a shows that TVD region as a shaded region. We see that the TVD region includes the situation where $\phi(\theta) = 0$, consistent with a first order upwind scheme. We wish to restrict our focus to second order schemes. For such schemes the slope that is chosen can range all the way from the right slope, which yields the Lax-Wendroff scheme, to the left slope, which yields the Beam-Warming scheme. I.e. this range is guaranteed to capture the full range

of second order schemes. The Lax-Wendroff scheme corresponds to making the choice $\phi(\theta)=1$ in eqn. (3.18) and the Beam-Warming scheme to the choice $\phi(\theta)=\theta$ in the same equation. These two choices are shown by the two dark solid lines in Fig. 3.7b. A second order TVD preserving scheme should lie between these two choices in addition to lying in the region shown in Fig. 3.7a. Thus a second order accurate TVD preserving scheme should lie in the shaded region shown in Fig. 3.7b. Notice that this also ensures $\phi(1)=1$ which was our further condition for second order accuracy. I.e. for situations where $\theta=1$ our scheme should revert back to the well-centered second order accurate Lax-Wendroff scheme. A final desirable attribute of TVD preserving second order accurate limiters is that they should produce a symmetric reconstruction of a symmetric mesh function. This property would help in reducing the upwind bias as much as possible; see Fig. 3.6b for an example of a very slight upwind bias. It can be shown that this symmetry condition is ensured by requiring

$$\phi(1/\theta)=\frac{\phi(\theta)}{\theta} \tag{3.24}$$

for all positive values of θ. Problem 3.6 at the end of this chapter steps the reader through this demonstration. All the limiters presented in this chapter satisfy the above symmetry condition. The shaded region shown in Fig. 3.7b is sometimes referred to as the *Sweby region*.

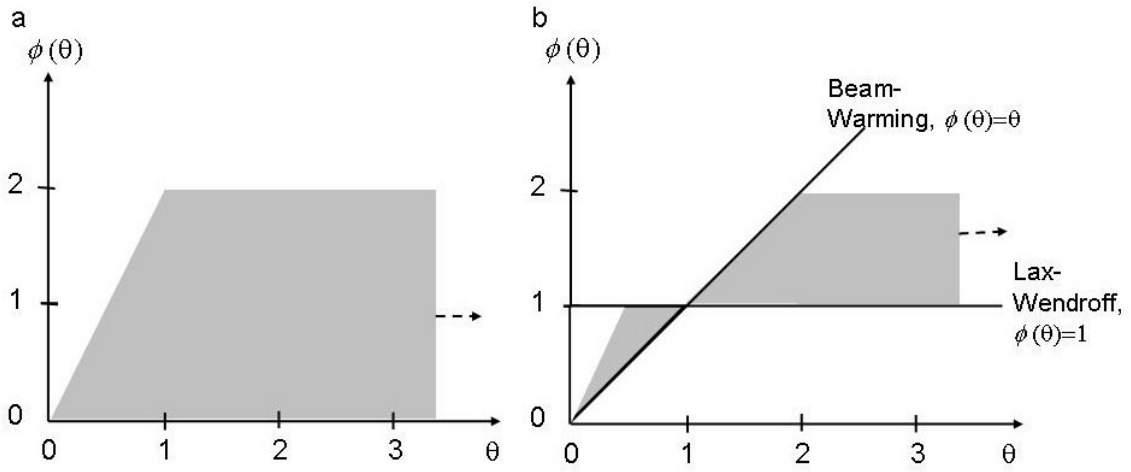


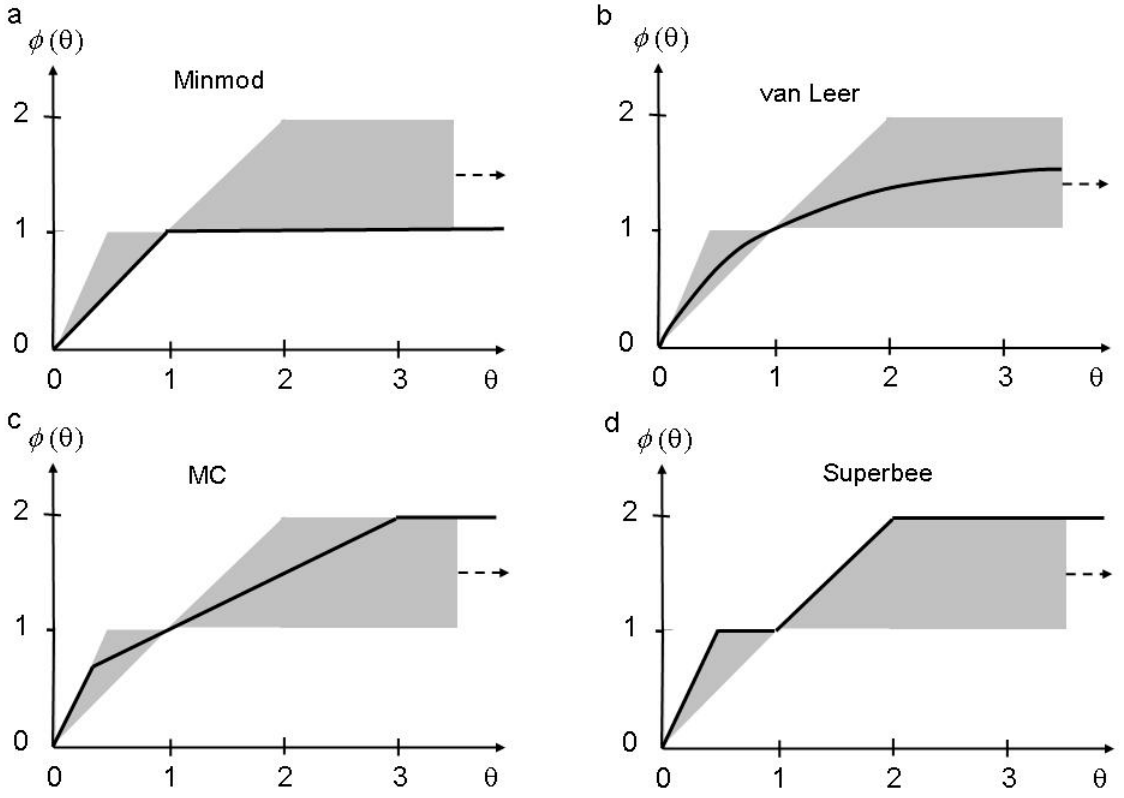
Fig 3.7a shows the entire TVD region, which is shaded in the figure. The shaded region in Fig. 3.7b shows the TVD region for second order accurate schemes. The second order TVD region should be bounded by the Lax-Wendroff and Beam-Warming schemes, which are also shown. The rightward pointing dashed arrows indicate that the shaded region extends onwards to the right.

We can now plot out the limiter functions for the four distinct types of popular, second order accurate limiters that were catalogued at the end of the last section. Written as flux limiters they become

$$\begin{aligned}
&\text{minmod} \;: \phi(\theta) = \text{minmod}(\,1,\ \theta) \\
&\text{van Leer} \;: \phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|} \\
&\text{MC} \qquad : \phi(\theta) = \max\left(0,\ \min\left(\,\frac{1}{2}(1+\theta),\ 2,\ 2\theta\right)\right) \\
&\text{Superbee} \;: \phi(\theta) = \max\left(\,0,\ \min(\,1,\ 2\theta),\ \min(\,2,\ \theta)\right)
\end{aligned} \tag{3.25}$$

Eqn. (3.25) shows that it is easy to transcribe from a flux limiter to a slope limiter and vice versa. The position of the flux limiters relative to the second order TVD region is shown in Figs. 3.8a,b,c and d for the minmod, van Leer, MC and Superbee limiters respectively. We see right away that all the limiter functions lie within the Sweby region for second order accurate TVD schemes, consistent with the fact that they are indeed second order accurate limiters. We can see from Fig. 3.8a that the minmod limiter always

has the smallest values and, therefore, produces the smallest slopes in the fluid profiles that it reconstructs. Consequently, schemes that use the minmod limiter, while second order accurate, will suffer the most amount of dissipation. The MC and van Leer limiters lie in the middle of the Sweby region. They will, therefore, produce slopes that are larger than the minmod limiters. The Superbee limiter lies at the upper boundary of the Sweby region. Consequently, it produces the largest slopes and is also the most compressive of limiters, a fact that can sometimes act to its detriment.



Figs. 3.8a to 3.8d show the limiter functions for the minmod, van Leer, MC and Superbee limiters.

Limiters from a Different Viewpoint

The previous two sections have given us both a pictorial and a rigorous way of understanding limiters. There is a further algebraically motivated viewpoint for understanding limiters that is more succinct and might please some readers. It also gives us the very important and interesting perspective that the limiters can depend on the kind

of underlying scheme that is used, see Arora and Roe (1997) and Toro (2009). We consider the advection equation, $u_t + a\,u_x = 0$ with a>0. Let us consider a portion of the mesh function around the zone "*i*", so that we focus on the mesh function $\overline{u}^n \equiv \left\{ ..., \overline{u}_{i-1}^n, \overline{u}_i^n, \overline{u}_{i+1}^n, ... \right\}$ at time t^n. Let Δx be the uniform zone size, let Δt be the timestep and let μ be the Courant number. In order to build a second order scheme, we endow zone "*i*" with an undivided difference $\overline{\Delta u_i}^n$, while leaving its value open for now. When $\overline{u}_{i-1}^n$, $\overline{u}_i^n$ and $\overline{u}_{i+1}^n$ are not monotone, i.e. when the value of $\overline{u}_i^n$ does not lie between $\overline{u}_{i-1}^n$ and $\overline{u}_{i+1}^n$, we can use our experience from the previous two sections to set $\overline{\Delta u_i}^n$ to zero. Thus we only consider situations where $\overline{u}_{i-1}^n \le \overline{u}_i^n \le \overline{u}_{i+1}^n$ or situations where $\overline{u}_{i-1}^n \ge \overline{u}_i^n \ge \overline{u}_{i+1}^n$ below.

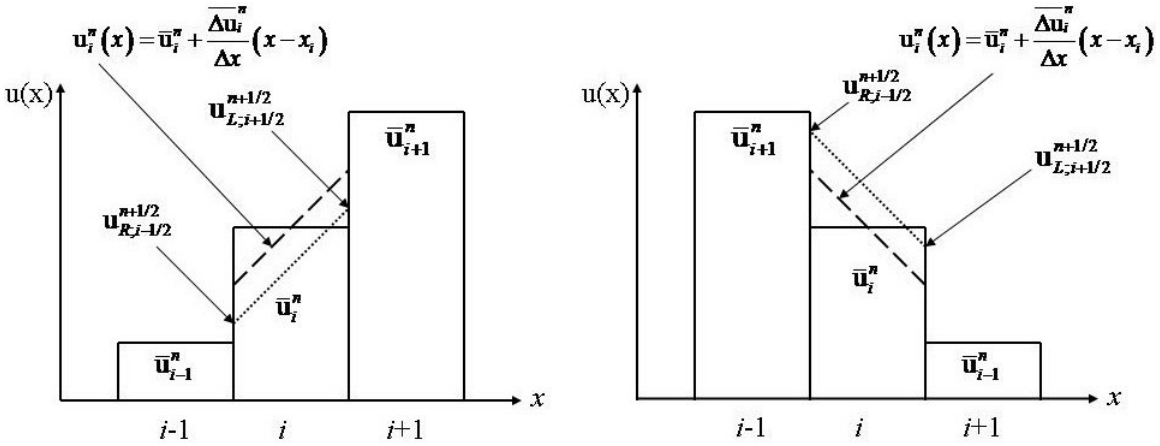
Within the zone "*i*" the time rate of update, u_t, can be discretized as $-a\,\overline{\Delta u_i}^n / \Delta x$. Defining $u_{L;i+1/2}^{n+1/2}$ to be the value of the solution at the left of the zone boundary "*i*+1/2" at a time $t^n + \Delta t/2$ we get

$$u_{L;i+1/2}^{n+1/2} = \overline{u}_i^n + \frac{1}{2}\overline{\Delta u_i}^n - \frac{\Delta t}{2}\, a\, \frac{\overline{\Delta u_i}^n}{\Delta x} = \overline{u}_i^n + \frac{1}{2}(1-\mu)\overline{\Delta u_i}^n$$

Notice that the flux in eqn. (3.6) can be written as $\overline{f}_{i+1/2}^{n+1/2} = a\,u_{L;i+1/2}^{n+1/2}$ which shows us that limiting $u_{L;i+1/2}^{n+1/2}$ and $u_{R;i-1/2}^{n+1/2}$ is tantamount to limiting the fluxes. Similarly, by defining $u_{R;i-1/2}^{n+1/2}$ to be the value of the solution at the right of the zone boundary "*i*–1/2" at a time $t^n + \Delta t/2$, we get

$$u_{R;i-1/2}^{n+1/2} = \overline{u}_i^n - \frac{1}{2}\overline{\Delta u_i}^n - \frac{\Delta t}{2}\, a\, \frac{\overline{\Delta u_i}^n}{\Delta x} = \overline{u}_i^n - \frac{1}{2}(1+\mu)\overline{\Delta u_i}^n$$

The monotonicity preserving property of the scheme can be formulated in terms of requiring $u_{R;i-1/2}^{n+1/2}$ to lie between $\bar{u}_{i-1}^{n}$ and $\bar{u}_{i}^{n}$ and, furthermore, by requiring $u_{L;i+1/2}^{n+1/2}$ to lie between $\bar{u}_{i}^{n}$ and $\bar{u}_{i+1}^{n}$. See the figure below.



The panel to the left shows a mesh function that is not decreasing with x. The panel to the right shows a mesh function that is not increasing with x. We solve the advection equation $u_t + a\,u_x = 0$ *with* $a>0$. *In each case the dashed line shows the piecewise linear reconstructed profile in x at time* t^n *while the dotted line shows the same profile at time* $t^n + \Delta t/2$.

Consider the case $\bar{u}_{i-1}^{n} \le \bar{u}_{i}^{n} \le \bar{u}_{i+1}^{n}$ so that the mesh function does not decrease with increasing "*x*". The left panel in the above figure provides an example. We wish to have $\bar{u}_{i-1}^{n} \le u_{R;i-1/2}^{n+1/2} \le \bar{u}_{i}^{n}$ which gives us

$$\bar{u}_{i-1}^{n} \le \bar{u}_{i}^{n} - \frac{1}{2}(1+\mu)\overline{\Delta u}_{i}^{n} \le \bar{u}_{i}^{n}$$

The two inequalities in the previous equation then yield the requirements that

$$0 \le \overline{\Delta u}_{i}^{n} \le \frac{2}{1+\mu}\left(\bar{u}_{i}^{n} - \bar{u}_{i-1}^{n}\right)$$

Similarly requiring that $\bar{u}_{i}^{n} \le u_{L;i+1/2}^{n+1/2} \le \bar{u}_{i+1}^{n}$ and simplifying gives us the requirements

$$0 \le \overline{\Delta u_i}^n \le \frac{2}{1-\mu}\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)$$

The above two sets of requirements can be amalgamated to yield

$$0 \le \overline{\Delta u_i}^n \le \min\left(\frac{2}{1+\mu}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right),\ \frac{2}{1-\mu}\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)\right)$$

Now consider the case $\bar{u}_{i-1}^n \ge \bar{u}_i^n \ge \bar{u}_{i+1}^n$ so that the mesh function does not increase with increasing "*x*". The right panel in the above figure provides an example. We wish to have $\bar{u}_{i-1}^n \ge u_{R;i-1/2}^{n+1/2} \ge \bar{u}_i^n$ which, after some simplification, gives us the conditions

$$-\frac{2}{1+\mu}\left(\bar{u}_{i-1}^n - \bar{u}_i^n\right) \le \overline{\Delta u_i}^n \le 0$$

Similarly, requiring $\bar{u}_i^n \ge u_{L;i+1/2}^{n+1/2} \ge \bar{u}_{i+1}^n$ and carrying out some simplifications, yields the conditions

$$-\frac{2}{1-\mu}\left(\bar{u}_i^n - \bar{u}_{i+1}^n\right) \le \overline{\Delta u_i}^n \le 0$$

Taken together, the above two sets of conditions yield

$$-\min\left(\frac{2}{1+\mu}\left(\bar{u}_{i-1}^n - \bar{u}_i^n\right),\ \frac{2}{1-\mu}\left(\bar{u}_i^n - \bar{u}_{i+1}^n\right)\right) \le \overline{\Delta u_i}^n \le 0$$

Putting all the conditions together for increasing and decreasing mesh functions together we get

$$\overline{\Delta u}_i^n = \frac{1}{2}\left(\text{sgn}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) + \text{sgn}\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)\right) \min\left(\frac{2}{1+\mu}\left|\bar{u}_i^n - \bar{u}_{i-1}^n\right|,\ \frac{2}{1-\mu}\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right|\right)$$

In most situations, as with the MC limiter that we have seen before, it helps to give the limiter the option of approaching a central slope so that we may also write

$$\overline{\Delta u}_i^n = \frac{1}{2}\left(\text{sgn}\left(\bar{u}_i^n - \bar{u}_{i-1}^n\right) + \text{sgn}\left(\bar{u}_{i+1}^n - \bar{u}_i^n\right)\right) \min\left(\frac{1}{2}\left|\bar{u}_{i+1}^n - \bar{u}_{i-1}^n\right|,\ \frac{2}{1+\mu}\left|\bar{u}_i^n - \bar{u}_{i-1}^n\right|,\ \frac{2}{1-\mu}\left|\bar{u}_{i+1}^n - \bar{u}_i^n\right|\right)$$

Notice that the limiters obtained via this process are dependent on the Courant number μ. Furthermore, the left and right undivided differences carry different weights with the result that the above two limiters only hold when a>0. For a<0, the weights $2/(1+\mu)$ and $2/(1-\mu)$ should switch places in the above two limiters. The limiters from this box do provide crisper profiles in several situations. However, it is also worth mentioning that Arora and Roe (1997) found that the Courant number-dependent limiters may not be as robust in certain circumstances as the ones we have studied previously. This limits their utility. For general hyperbolic systems, they also require us to carry out a characteristic decomposition, which we study in the next section. The characteristic decomposition of large hyperbolic systems can be computationally costly.

The van Albada Limiter

The Sweby region gives us a formal way of analyzing and categorizing second order accurate limiters. In practice, limiters might be constructed that serve different purposes. Several practical limiters have been constructed which lie quite close to the Sweby region, even though they do not lie entirely within it. A prominent example is the van Albada limiter (van Albada *et al.* 1982) given by

$$\text{vanAlbada}\ (a,b) = \frac{a^2 b + b^2 a}{a^2 + b^2 + \varepsilon}$$

Here ε is a tiny number designed to prevent division by zero. The van Albada limiter can, of course, be brought into the Sweby region by prefixing the right hand side with $\frac{1}{2}\left(\text{sgn}\left(a\right)+\text{sgn}\left(b\right)\right)$, but it also works well as given. Even when the slopes have opposite signs the limiter picks out the smaller of the two slopes, thus stabilizing the numerical method. While this limiter doesn't produce the sharpest profiles for explicit advection schemes, it is very useful for implicit formulations. Because of its smoothness, it produces a flux that has a smooth and differentiable dependence on the mesh function. This smooth variation is very useful for the convergence of iterative linear algebra methods.

3.4) Linear Hyperbolic Systems and the Riemann Problem

We split this section into several connected parts. In Sub-section 3.4.1 we discuss the solution of linear hyperbolic equations for continuous, once-differentiable initial conditions. Sub-section 3.4.2 discusses discontinuous initial conditions and introduces the Riemann problem. Sub-section 3.4.3 shows the importance of the Riemann problem as an essential building block for numerical solutions of hyperbolic systems. Sub-section 3.4.4 explains how the Riemann problem restores stability to a second order accurate TVD scheme for linear hyperbolic systems while retaining consistency of the numerical fluxes that it provides to a numerical scheme. Sub-section 3.4.5 presents the fluctuation form which will be very useful in our subsequent study of non-conservative systems. The box presented at the end of this section instantiates the mathematics in this section for the linearized one-dimensional Euler equations. It might be helpful to read it in conjunction with this section so that the reader can see the arguments instantiated for a particular physical system. The reader is also requested to review Section 1.5 before reading this section.

3.4.1) Solution of Linear Hyperbolic PDEs for Continuous Initial Data

Many of the systems of conservation laws that interest us have a predominantly hyperbolic component with an additional non-ideal component that may be parabolic. Here we focus on the hyperbolic part, ignoring non-ideal contributions. For now we further restrict attention to one dimensional variations. The conservation law can then be formally written as $U_t + F(U)_x = 0$ where the vectors U and F are column vectors with "*M*" components. As seen in Chapter 1, the conservation law can be linearized to yield an equation of the form

$$U_t + A\, U_x = 0 \tag{3.26}$$

where "A" is an $M \times M$ matrix that is sometimes referred to as the *characteristic matrix*. We say that the system is hyperbolic if "A" has "*M*" real eigenvalues, $\lambda^1, \lambda^2, ..., \lambda^M$. Such a hyperbolic system is sometimes referred to as an $M \times M$ hyperbolic system. Physically, as we have demonstrated in Sub-sections 1.5.2 and 1.5.3, it means that a very small one-dimensional disturbance produced at any location will propagate away from that location as "*M*" independent waves along the x-axis. In the next paragraph we study the role of these eigenvalues in characterizing a hyperbolic system.

For certain hyperbolic systems the eigenvalues can be arranged in an ordered sequence from smallest to largest, i.e. $\lambda^1 < \lambda^2 < ... < \lambda^M$. Furthermore, if that order is preserved regardless of the value of "U" we call the hyperbolic system *strictly hyperbolic*. Physically, the waves emanating from a small perturbation to the constant state of a strictly hyperbolic system are guaranteed to always have the same ordering in space-time. Because the waves in a strictly hyperbolic system are well-separated, elegant theorems can be proved to ensure that solution techniques can be found, see Lax (1972). Typical hyperbolic systems of interest to us are not strictly hyperbolic; consider the three-dimensional Euler equations as an example. For the 5×5 Euler system we have $\lambda^1 < \lambda^2 \le \lambda^3 \le \lambda^4 < \lambda^5$ where the extremal wave families correspond to the left- and right-going sound waves and the central three correspond to the entropy wave and the two transverse shear waves. We call such systems *non-strictly hyperbolic*. Fortunately for

the Euler system, the *degenerate eigenvalues* correspond to *linearly degenerate wave families* and a result by Lax comes to the rescue ensuring that reliable numerical solution methods can be found in certain instances. We will learn about linearly degenerate wave families later. Unfortunately, several of the other hyperbolic systems of interest to us, such as two-phase flow, non-linear elasticity, MHD and relativistic MHD, have characteristic matrices with worse properties and the intrepid computationalist has to persevere with the situation as it is.

In general, the matrix "A" will be a strongly non-linear function of the solution vector. We will study such situations later. A much simpler situation arises when the matrix "A" is a constant. The solution for such systems is easily found and is catalogued in this section. The ideas developed here will also be useful for the numerical treatment of non-linear hyperbolic systems. In particular, even when one is dealing with a non-linear hyperbolic system, one can always achieve the form in eqn. (3.26) by freezing the matrix "A" at any location consistent with the local values of the solution vector "U". Consequently, while we restrict our attention to linear hyperbolic systems in this section, we will also gain important early insights into treating the non-linear case.

Associated with each of the "M" eigenvalues , λ^m with m=1,…M , we can write an equation for the right and left eigenvectors as

$$\mathrm{A}\, r^m = \lambda^m\, r^m \;; \quad l^m\, \mathrm{A} = \lambda^m\, l^m \tag{3.27}$$

where r^m is a column vector with "M" components and l^m is a row vector with "M" components. We assume that the eigenvalues form an ordered set, $\lambda^1 < \lambda^2 < ... < \lambda^M$. The eigenvectors can be arranged in the same order so that we can write a matrix R whose m$^{\text{th}}$ column is r^m. Similarly, we can obtain a matrix of left eigenvectors L whose m$^{\text{th}}$ row is l^m , see Fig. 1.11. Notice that we set $L = R^{-1}$ so that the left and right eigenvectors are orthonormal relative to eachother, i.e. $l^m\, r^n = \delta_{mn}$ where δ_{mn} is the Kronecker delta function. Defining the diagonal matrix $\Lambda = \mathrm{diag}\left\{\lambda^1, \lambda^2, ..., \lambda^M\right\}$ we can write

$$L\,\mathrm{A}\,R = \Lambda \quad \text{or} \quad \mathrm{A} = R\,\Lambda\,L \tag{3.28}$$

By left-multiplying eqn. (3.26) with the m^{th} left eigenvector l^m we get

$$l^m\,\mathrm{U}_t + l^m\,\mathrm{A}\,\mathrm{U}_x = w_t^m + \lambda^m\,l^m\,\mathrm{U}_x = w_t^m + \lambda^m\,w_x^m = 0 \quad \text{where } w^m \equiv l^m\,\mathrm{U} \tag{3.29}$$

We call the scalar variable $w^m \equiv l^m\,\mathrm{U}$ the *characteristic variable* or *eigenweight* for the m^{th} wave family. Eqn. (3.29) shows us that the characteristic variable for the m^{th} wave family satisfies the simple advection equation $w_t^m + \lambda^m\,w_x^m = 0$. In other words, the eigenweight w^m propagates quite simply as a wave with the speed λ^m. This process of decomposing a general hyperbolic system $\mathrm{U}_t + \mathrm{A}\,\mathrm{U}_x = 0$ into a set of simple advection equations $w_t^m + \lambda^m\,w_x^m = 0$ in the characteristic variables is called *characteristic decomposition*. It is important to also point out that many aspects of the mathematics that were used in eqn. (3.29) would not hold exactly for non-linear systems. In particular, writing $l^m\,\mathrm{U}_t = \left(l^m\,\mathrm{U}\right)_t = w_t^m$ assumes that l^m is a constant in space and time. Similarly, for a linear system the waves from eqn. (3.29) can propagate past each other without a change in their form, i.e. without dispersion, a fact that does not hold true for non-linear hyperbolic systems. However, eqn. (3.29) is so important in developing our understanding of hyperbolic systems that we will later on be cavalier enough to rely on it even for non-linear hyperbolic systems. When it comes to designing numerical schemes for hyperbolic systems, we simply do not have any substantially better perspective to hang our hats on.

Eqn. (3.29) has shown us that the parts of the solution vector that are projected into characteristic variables travel with the wave speed associated with that variable. The *Cauchy problem for hyperbolic systems* tells us that if the solution is defined in a smooth and differentiable way on a non-characteristic surface in space-time then it can indeed be evolved further for a finite distance in time. For our purposes we will assume that the

initial conditions are given at time $t = 0$ on the *x*-axis by an *M*-component vector $\mathrm{U}_0(x)$ each of whose components are smooth and differentiable at least once. We can then obtain "*M*" smooth functions given by

$$w_0^m(x) = l^m \, \mathrm{U}_0(x) \qquad \text{for } m = 1, ..., M \tag{3.30}$$

Since $\mathrm{U}_0(x)$ is a specified function of *x*, eqn. (3.30) tells us that $w_0^m(x)$ is a known function of *x*. In light of eqn. (3.29) which says that the m^{th} characteristic variable obeys the advection equation $w_t^m + \lambda^m \, w_x^m = 0$, the solution for $t > 0$ is given by

$$\mathrm{U}(x,t) = \sum_{m=1}^{M} w_0^m\left(x - \lambda^m t\right) r^m \tag{3.31}$$

It is easy to substitute eqn. (3.31) into eqn. (3.26) and verify that it is the right solution of the linear PDE that satisfies the initial conditions. Thus if all we wanted to do was solve a smooth problem with initial condition $\mathrm{U}_0(x)$ over the entire *x*-axis then we would only have to generate the profiles $w_0^m(x)$ for each characteristic field, advect (i.e. shift) them around with the appropriate wave speed to get $w_0^m\left(x - \lambda^m t\right)$ and then use eqn. (3.31) to obtain the solution at some later time. In practice, we would also have to be mindful of boundary conditions on a finite domain. But it is worth pointing out that the solution of the linear problem with smooth initial conditions is rather simple. However, linear hyperbolic PDEs also admit discontinuous solutions and we study those solutions next.

3.4.2) Solution of Linear Hyperbolic PDEs for Discontinuous Initial Data: Simple Waves and the Riemann Problem

The previous sub-section showed us how to obtain the solution at all times $t > 0$ when the initial conditions are specified as a continuous function $\mathrm{U}_0(x)$ at time $t = 0$.

When $U_0(x)$ is continuous and differentiable, there is only an infinitesimal jump from one point to an immediately neighboring one. It turns out that eqn. (3.26) also admits discontinuous initial conditions. In this sub-section we will study the evolution of discontinuous initial data in two gradual stages. In the first stage, we study simple waves where the jump in the initial data has to have a specific form. In the second stage of our study, we consider the Riemann problem where one can have an arbitrary change in initial conditions at a point of discontinuity.

Let us first study simple waves. The differential form of the hyperbolic PDE in eqn. (3.26) cannot, by itself, represent a discontinuity because the derivatives would be ill-defined at the discontinuity. However, there is an integral sense in which one can justify the presence of discontinuous solutions. Thus imagine a situation where we have $U_0(x) = U_L$ for $x<0$ and $U_0(x) = U_R$ for $x \geq 0$. Let that discontinuity propagate from the origin in the x-direction so that at a time "T" it has propagated a distance "X" as shown in Fig. 3.9. We can now integrate $U_t + A\,U_x = 0$ over the rectangle $[0,X]\times[0,T]$ in space and time, see Fig. 3.9. Using integration by parts, which we illustrate below, we get

$$\int_{t=0}^{t=T}\int_{x=0}^{x=X}\left(U_t + A\,U_x\right)dx\,dt = 0 \quad \Leftrightarrow \quad \left[\int_{x=0}^{x=X} U\,dx\right]_{t=0}^{t=T} + A\left[\int_{t=0}^{t=T} U\,dt\right]_{x=0}^{x=X} = 0$$
$$\Leftrightarrow \quad U_L\,X - U_R\,X + A\left(U_R\,T - U_L\,T\right) = 0 \quad \Leftrightarrow \quad A\left(U_R - U_L\right) = \frac{X}{T}\left(U_R - U_L\right) \tag{3.32}$$

We then see from eqn. (3.32) that in order for a discontinuity to satisfy the integral form of the linear hyperbolic system, it must move with a speed given by λ which satisfies the equation

$$A\left(U_R - U_L\right) = \lambda\left(U_R - U_L\right) \tag{3.33}$$

Eqn. (3.33) shows us that the speed of the discontinuity must be one of the eigenvalues of the characteristic matrix "A". Furthermore, the jump $U_R - U_L$ must be proportional to the corresponding eigenvector of "A". Eqn. (3.33) is a special form of the *Rankine-Hugoniot jump conditions* which we will study even further for non-linear hyperbolic systems in the next two chapters. We, therefore, see that the eigenvectors are not just useful because they allow us to propagate smooth solutions, as was done in eqn. (3.31), but they are also useful in propagating discontinuous solutions to eqn. (3.26). Such discontinuous solutions of hyperbolic systems that satisfy eqn. (3.32), which holds only in an integral sense, are referred to as *weak solutions* of the hyperbolic equations. In the next chapter we will see that nonlinear hyperbolic systems that can be cast in conservation form can also support weak solutions. The existence of a conservation form is crucial, because if the hyperbolic PDE does not have a conservation form then we cannot demonstrate the existence of weak solutions.

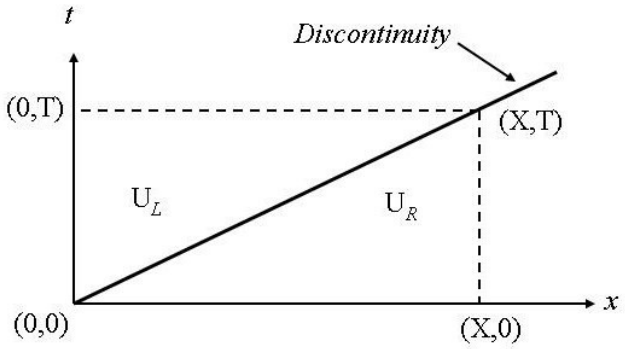


Fig. 3.9 shows the propagating discontinuity in space-time. It propagates a distance X *along the x-axis in a time* T. *The linear hyperbolic equation is integrated over the rectangle shown by the dashed line.*

The simplest form of discontinuous solution admitted by eqn. (3.26), therefore, consists of a *simple wave*. Such a solution can be defined for any of the "*M*" wave families of eqn. (3.26). The simple wave for the m^{th} wave family that is centered at the origin $x = 0$ consists of using the initial conditions

$$U_0(x) = U_L \qquad \text{for } x<0 \quad ; U_0(x) = U_R = U_L + \alpha^m r^m \qquad \text{for } x \geq 0 \tag{3.34}$$

Consequently, U_L and U_R correspond to the solutions to the left and right of the discontinuity respectively. α^m is a constant in eqn. (3.34) and is usually referred to as an eigenweight of the eivenvector r^m. The time-evolution of this simple wave is given by

$$\begin{aligned} \mathrm{U}(x,t) &= \mathrm{U}_L && \text{for } x<\lambda^m\ t \\ &= \mathrm{U}_R = \mathrm{U}_L + \alpha^m\ r^m && \text{for } x \geq \lambda^m\ t \end{aligned} \tag{3.35}$$

I.e. to the left of the characteristic curve $x = \lambda^m\ t$ the solution is U_L, to the right of it the solution is U_R. We have the freedom to specify U_L or U_R, but not both. Notice that the jump between U_L and U_R for a simple wave has to be carefully arranged so that it is exactly equal to some scalar multiple of the m^{th} right eigenvector, i.e. $\alpha^m\ r^m$.

Having studied simple waves of finite amplitude, it is very natural to ask, what happens if the left and right states, i.e. U_L and U_R respectively, are chosen randomly? Let us assume, as before, that the discontinuity is located at $x = 0$ so that the initial conditions at time $t = 0$ are given by $\mathrm{U}_0(x) = \mathrm{U}_L$ for $x < 0$ and $\mathrm{U}_0(x) = \mathrm{U}_R$ for $x \geq 0$. Clearly, if the difference between the two states is very small then the dispersion analysis for linearized PDEs from Chapter 1 tells us that we should expect "*M*" waves with very small wave strength to propagate away from the origin, see Fig. 1.9. If the jump $\mathrm{U}_R - \mathrm{U}_L$ is of finite amplitude and the hyperbolic system is linear then a similar result prevails. Thus, a set of "*M*" waves of finite amplitude propagate away from the origin. Fig. 3.10 shows a schematic diagram of this situation. Eqns. (3.34) and (3.35) give us a very useful hint that the difference in the right and left states, $\mathrm{U}_R - \mathrm{U}_L$, should somehow be related to the right eigenvectors. To make that relationship more concrete, we now remind ourselves of a result from matrix methods which says that any real vector with "*M*" components can be projected onto the eigenvectors of a square matrix "A" if that matrix has real and disjoint eigenvalues. To use the terminology of vector spaces, the right eigenspace of the matrix "A" in eqn. (3.26) is complete because it has real and disjoint

eigenvalues. As a result, any real M-component vector, say $\mathrm{U}_R - \mathrm{U}_L$, can be projected onto the eigenspace $\left\{r^1, r^2, ..., r^M\right\}$. Thus we can write

$$\mathrm{U}_R - \mathrm{U}_L = \sum_{m=1}^{M} \alpha^m \, r^m \quad \text{where} \quad \alpha^m \equiv l^m \left(\mathrm{U}_R - \mathrm{U}_L\right) \tag{3.36}$$

From eqn. (3.36) we see that the set of left eigenvectors $\left\{l^1, l^2, ..., l^M\right\}$ enable us to project the vector $\mathrm{U}_R - \mathrm{U}_L$ into the vector space of the right eigenvectors. Notice that if we apply the construction of eqn. (3.36) to the left and right states of the simple wave for the m^{th} wave family, i.e. if U_L and U_R are given by eqn. (3.34), then we clearly make the identification that α^m represents the eigenweight of the single eigenvector r^m. Operationally, we obtain the eigenweights α^m by observing that $\alpha^m \equiv l^m \left(\mathrm{U}_R - \mathrm{U}_L\right)$. We urge the reader to see the analogy between eqn. (3.36) which gives the eigenweights for discontinuous initial conditions and eqn. (3.30) for obtaining the eigenweights when we have smooth initial conditions. This realization now enables us to interpret the eigenweights and the solution given in eqn. (3.36). Taking our cue from eqn. (3.35), we realize that each jump by $\alpha^m r^m$ introduces another simple wave and eqn. (3.36) decomposes the general jump $\mathrm{U}_R - \mathrm{U}_L$ into a sequence of simple waves, each moving with their characteristic speed λ^m. Interpreting Fig. 3.10 we see that we have a set of $M-1$ constant states $\left\{\mathrm{U}^{(1)}, \mathrm{U}^{(2)}, ..., \mathrm{U}^{(m)}, ..., \mathrm{U}^{(M-1)}\right\}$ between U_L and U_R where the m^{th} constant state, $\mathrm{U}^{(m)}$, lies in the region bounded by $\lambda^m \, t < x < \lambda^{m+1} \, t$. In other words, the m^{th} constant state, $\mathrm{U}^{(m)}$, lies between the m^{th} characteristic surface and the $(m+1)^{\text{th}}$ characteristic surface. The solution at all points in space and time is then given by

$$
\begin{aligned}
\mathrm{U}(x,t) &= \mathrm{U}_L && \text{for } \frac{x}{t} < \lambda^1 \\
&= \mathrm{U}^{(m)} \equiv \mathrm{U}_L + \sum_{p=1}^{m} \alpha^p \, r^p = \mathrm{U}_R - \sum_{p=m+1}^{M} \alpha^p \, r^p && \text{for } \lambda^m < \frac{x}{t} < \lambda^{m+1} \;,\; m = 1,\ldots,M-1 \\
&= \mathrm{U}_R && \text{for } \lambda^M < \frac{x}{t}
\end{aligned}
\tag{3.37}
$$

Notice from Fig. 3.10 that the jump between $\mathrm{U}^{(m-1)}$ and $\mathrm{U}^{(m)}$ is mediated by the characteristic surface $x = \lambda^m \, t$ associated with the m^{th} wave. Eqn. (3.37) clearly shows that the jump $\mathrm{U}^{(m)} - \mathrm{U}^{(m-1)}$ is then proportional to the right eigenvector r^m . Thus the general jump $\mathrm{U}_R - \mathrm{U}_L$ is resolved by *fitting* a sequence of "*M*" simple waves between U_L and U_R. We urge the reader to see the analogy between eqn. (3.37) which gives the solution for discontinuous initial conditions and eqn. (3.31) for obtaining the solution when we have smooth initial conditions.

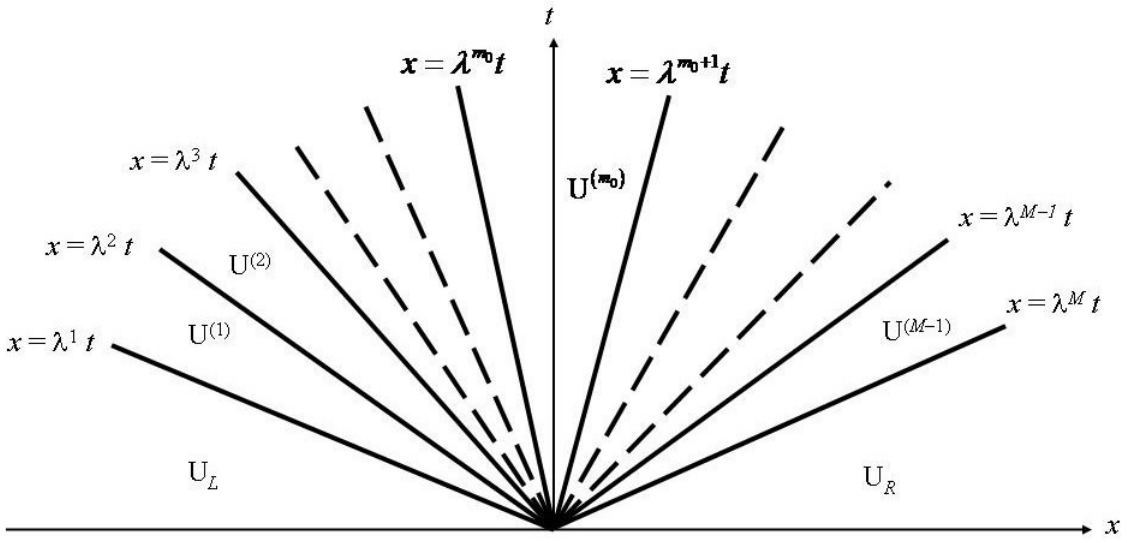


Fig. 3.10 shows the space-time diagram for the propagation of finite amplitude (or infinitesimal) perturbations for an M-component linear hyperbolic system. The solid lines show waves; the dashed lines represent the presence of further waves that may not be explicitly shown here. The left and right states are denoted by U_L *and* U_R. *The resolved state of the Riemann problem is shown as* $\mathbf{U}^{(RS)} \equiv \mathbf{U}^{(m_0)}$.

Observe from eqn. (3.37) that $\mathrm{U}^{(m)}$ can be obtained by starting from U_L and sequentially adding in the contributions of the right eigenvectors $\alpha^p r^p$ with $p \le m$. Thus envision traversing Fig. 3.10 from left to right. Crossing the characteristic curve associated with each simple wave in Fig. 3.10 adds the contribution from that simple wave to the solution. Similarly, $\mathrm{U}^{(m)}$ can also be obtained by starting from U_R and sequentially subtracting off the contributions of the right eigenvectors $\alpha^p r^p$ with $p \ge m+1$. This is tantamount to traversing Fig. 3.10 from right to left. All this is, of course, consistent with our interpretation of eqn. (3.36). Notice too that eqn. (3.37) is a *similarity solution* because it only depends on the similarity variable x/t . In physical terms, the characteristics that carry the fluctuations in the jump $\mathrm{U}_R - \mathrm{U}_L$ are straight lines in space-time, hence the solution is self-similar. The solution at a later time just looks like an expanded version of the solution at an earlier time. Thus the problem where U_L and U_R are arbitrarily specified for a linear hyperbolic system gives rise to a system of "*M*" simple waves with *M*–1 constant states lying between U_L and U_R. This problem is called the *Riemann problem* in honor of Bernhardt Riemann who understood its importance for linear as well as non-linear hyperbolic systems, Riemann (1860). Because of its fan-like structure, Fig. 3.10 is also referred to as the *Riemann fan*. The Riemann problem is an important building block for solution techniques for linear as well as non-linear hyperbolic systems.

3.4.3) The Riemann Problem as a Building Block for the Numerical Solution of Hyperbolic Systems

We now demonstrate why the Riemann problem is an important building block for the numerical solution of hyperbolic problems. Figs. 3.1 and 3.4 show us two possible monotonicity preserving reconstruction strategies with piecewise constant and piecewise linear reconstruction respectively. By looking at Figs. 3.1 and 3.4 we can easily realize that any monotonicity preserving reconstruction of physical variables results in discontinuities at zone boundaries. This is true regardless of whether the slabs are

piecewise constant, as they are in Godunov's original method, or whether they have piecewise linear (or piecewise parabolic, or piecewise cubic…) profiles. A higher order reconstruction only reduces the magnitude of the jumps at the zone boundaries when the mesh function is smooth. In doing so, it reduces the amount of dissipation introduced by the scheme. However, for arbitrary mesh functions, the jumps cannot be eliminated if one wants a monotonicity preserving reconstruction of the physical variables. The Riemann problem, which is the object or our study, then allows us to obtain a physically consistent strategy for following the evolution of the jumps at the zone boundaries.

Fig. 3.11 provides a schematic diagram showing a piecewise constant mesh function and its evolution in space-time. The upper panel in Fig. 3.11 shows the slabs of fluid along with the jumps at the zone boundaries. The jumps can have any value, so they result in a Riemann fan at each zone boundary. The upper panel in Fig. 3.11 is meant to be the analogue of Fig. 3.1b for the donor cell method. I.e. in this section we seek to construct an analogue of the donor cell method with its upwinded fluxes for linear hyperbolic systems. The lower panel in Fig. 3.11 shows how the waves from the Riemann problem at each zone boundary propagate away from that zone boundary. Notice too that we have found it very desirable to write the solution in flux form. Using the top panel Fig. 3.11, and making the identification $\mathrm{F} = \mathrm{A}\,\mathrm{U}$, we can write Godunov's first order scheme as

$$\overline{\mathrm{U}}_i^{n+1} = \overline{\mathrm{U}}_i^n - \frac{\Delta t}{\Delta x}\left(\overline{\mathrm{F}}_{i+1/2}^n - \overline{\mathrm{F}}_{i-1/2}^n\right) \tag{3.38}$$

Here the index "*i*" labels zone centers of a mesh of with mesh size Δx and eqn. (3.38) depicts a timestep from time t^n to $t^{n+1} = t^n + \Delta t$. The zone boundaries of zone "*i*" are labeled "*i*+1/2" and "*i*–1/2". Recall from eqns. (2.2) and (2.3) that the *numerical flux* $\overline{\mathrm{F}}_{i+1/2}^n$ is a time-average of the flux at the zone boundary. The goal is to find properly *upwinded* fluxes for eqn. (3.38). In other words, we wish to find fluxes that build in the realization that there are discontinuities in the solution and incorporate the fact that the discontinuity at each boundary will split into a family of simple waves that move in

different directions. More specifically, from Fig. 3.11 we see that there will always be some discontinuities in the solution vector at the zone boundaries. For example, at zone boundary "*i+1/2*" Fig. 3.11 shows us that we have a left state $\mathrm{U}_{L;i+1/2}$ and a right state $\mathrm{U}_{R;i+1/2}$ which are both used to form the upwinded flux $\overline{\mathrm{F}}^{n}_{i+1/2}$ in eqn. (3.38). Fig. 3.10 shows us that these jumps will result in a Riemann fan with simple waves propagating in different directions. The upwinded flux is the physically meaningful flux that is produced by the Riemann problem, which correctly resolves which waves flow to the left and which waves flow to the right of the original discontinuity. The self-similarity of the Riemann problem, as well as the fact that we restrict our attention in eqn. (3.38) to first order schemes in one dimension, makes it easy for us to pick out the flux that we need. We do that next.

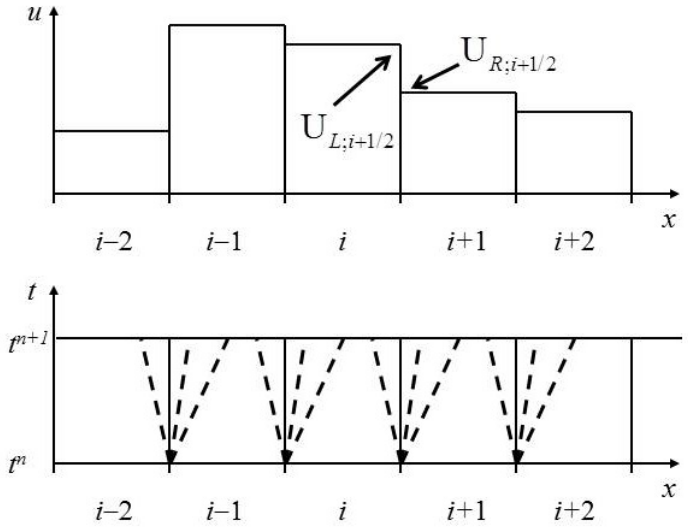


Fig. 3.11 schematically shows Godunov's method on a one-dimensional mesh. Here each zone is represented as a slab of fluid in the top panel. The bottom panel shows the evolution in space and time of the Riemann problems at zone boundaries. Because the problem is linear, the dashed lines show Riemann problems with an identical pattern of waves at each of the zone boundaries. The strength of the waves can, however, vary from one zone boundary to the next.

From the third panel in Fig. 3.11 we see that we want the solution of the Riemann problem at the zone boundaries. Our zone boundaries don't move in time. Let us therefore shift our coordinate system so that the origin, x=0, coincides with the zone boundary of interest. (As long as we avoid the intricacies of a mesh with moving boundaries, our simple use of non-moving boundaries will serve us well on several problems.) This is tantamount to asking for the solution of the Riemann problem at $x = 0$

for times $t > 0$ in Fig. 3.10. We, therefore, seek a state $\mathrm{U}^{(RS)} \equiv \mathrm{U}^{(m_0)}$ which overlies $x = 0$ as shown in Fig. 3.10. The two characteristics $x = \lambda^{m_0} t$ and $x = \lambda^{m_0+1} t$ straddle the zone boundary that we are interested in. This state is often referred to as the *resolved state of the Riemann problem*. To obtain the resolved state we have, therefore, to distinguish between waves that move to the right and waves that move to the left. For the resolved state we have

$$\begin{aligned} \mathrm{U}^{(RS)} \equiv \mathrm{U}^{(m_0)} &= \mathrm{U}_L && \text{if } 0 < \lambda^1 \\ &= \mathrm{U}_L + \sum_{p=1}^{m_0} \alpha^p \, r^p = \mathrm{U}_R - \sum_{p=m_0+1}^{M} \alpha^p \, r^p && \text{if } \lambda^{m_0} < 0 \le \lambda^{m_0+1} \\ &= \mathrm{U}_R && \text{if } \lambda^M \le 0 \end{aligned} \qquad (3.39)$$

The *resolved flux* is the flux that corresponds to the resolved state. It is also known as the *numerical flux*, because it is the flux that is evaluated at zone boundaries in a numerical code. For a linear hyperbolic system, the flux at any point in space-time is easily obtained by right multiplying the characteristic matrix "A" by the solution from eqn. (3.39). The resolved flux that we desire is just $\mathrm{F}^{(RS)} \equiv \mathrm{A}\,\mathrm{U}^{(RS)}$, though it can be a fair bit more complicated for a non-linear hyperbolic system. The numerical flux that we seek for eqn. (3.38) is just a space-time average of the resolved flux at the zone boundary. For the simple case of a first order scheme, that averaging is trivial.

Because eqn. (3.39) and the resolved flux that results from it are very important, a good bit of attention is lavished on obtaining compact, computationally efficient, expressions for it. Eqn. (3.39) is not very well-suited for computer implementation because it relies on analyzing the foliation of the waves in the Riemann problem. Our goal in this paragraph is to obtain expressions that automate this process and are, therefore, easy to implement in a computer code. Defining the auxiliary variables

$$\lambda^{+,m} \equiv \max\left(\lambda^m, 0\right) \quad ; \quad \lambda^{-,m} \equiv \min\left(\lambda^m, 0\right) \quad ; \quad \mathrm{F}_L \equiv \mathrm{A}\,\mathrm{U}_L \quad ; \quad \mathrm{F}_R \equiv \mathrm{A}\,\mathrm{U}_R \qquad (3.40)$$

and evaluating the desired numerical flux $\mathrm{A}\,\mathrm{U}^{(RS)}$ from eqn. (3.39) enables us to write the numerical flux in three equivalent forms as

$$\mathrm{F}^{(RS)} = \mathrm{F}_L + \sum_{m=1}^{M} \lambda^{-,m}\ \alpha^m\ r^m \tag{3.41a}$$

$$\mathrm{F}^{(RS)} = \mathrm{F}_R - \sum_{m=1}^{M} \lambda^{+,m}\ \alpha^m\ r^m \tag{3.41b}$$

$$\mathrm{F}^{(RS)} = \frac{1}{2}\left(\mathrm{F}_R + \mathrm{F}_L\right) - \frac{1}{2}\sum_{m=1}^{M} \left|\lambda^m\right|\ \alpha^m\ r^m \tag{3.41c}$$

All three expressions in eqn. (3.41) are equivalent. Observe that $\lambda^{+,m}$ and $\lambda^{-,m}$ in eqn. (3.40) are designed to make it easy for us to distinguish between wave families that move to the right and those that move to the left. Notice that $\lambda^{-,m} = 0$ for $m > m_0$ thus enabling us to justify the first expression in eqn. (3.41). Similarly, $\lambda^{+,m} = 0$ for $m \le m_0$, thus justifying the second expression in eqn. (3.41). The third expression in eqn. (3.41) is just an arithmetic average of the first two expressions in the same equation. Thus instead of identifying a state $\mathrm{U}^{(m_0)}$ and extending the summation selectively, as we do in eqn. (3.39), our definition of $\lambda^{+,m}$ and $\lambda^{-,m}$ enables us to be cavalier about the summation indices in eqn. (3.41). We put our new notation to work by showing that the computation of the resolved flux can be automated in order to facilitate implementation on a computer. Using $\alpha^m \equiv l^m\left(\mathrm{U}_R - \mathrm{U}_L\right)$ from eqn. (3.36) in the above equation then allows us to write the numerical flux as

$$\mathrm{F}^{(RS)} = \mathrm{F}_L + \mathrm{A}^{-}\left(\mathrm{U}_R - \mathrm{U}_L\right) \tag{3.42a}$$

$$\mathrm{F}^{(RS)} = \mathrm{F}_R - \mathrm{A}^{+}\left(\mathrm{U}_R - \mathrm{U}_L\right) \tag{3.42b}$$

$$\mathrm{F}^{(RS)} = \frac{1}{2}\left(\mathrm{F}_R + \mathrm{F}_L\right) - \frac{1}{2}\left|\mathrm{A}\right|\left(\mathrm{U}_R - \mathrm{U}_L\right) \tag{3.42c}$$

All three expressions in eqn. (3.42) are equivalent. The above equation is completed by defining the matrices

$$\begin{aligned} &\Lambda^{+} \equiv \mathrm{diag}\left\{\lambda^{+,1}, \lambda^{+,2}, ..., \lambda^{+,M}\right\} \quad ; \quad \Lambda^{-} \equiv \mathrm{diag}\left\{\lambda^{-,1}, \lambda^{-,2}, ..., \lambda^{-,M}\right\} \quad ; \\ &\left|\Lambda\right| \equiv \mathrm{diag}\left\{ \left|\lambda^{1}\right| , \left|\lambda^{2}\right| , ..., \left|\lambda^{M}\right| \right\} \quad ; \quad \mathrm{A}^{+} \equiv R\Lambda^{+}L \quad ; \quad \mathrm{A}^{-} \equiv R\Lambda^{-}L \quad ; \quad \left|\mathrm{A}\right| \equiv R\left|\Lambda\right|L \end{aligned} \tag{3.43}$$

Notice that A^{+}, A^{-} and $\left|\mathrm{A}\right|$ in eqn. (3.43) have definitions that parallel that of "A" in eqn. (3.28). To demonstrate that the expressions in eqn. (3.42) are equivalent to those in eqn. (3.41) it is easiest to substitute the eqns. (3.43) in the expressions given for the resolved flux by eqn. (3.42). For a linear hyperbolic system, the matrices A^{+}, A^{-} and $\left|\mathrm{A}\right|$ are computed once and for all using eqn. (3.43). Any of the expressions in eqn. (3.42) then provide an automatic evaluation of the resolved flux in a form that is suitable for implementation in a computer code.

3.4.4) Consistency and Dissipation Properties of the Linear Riemann Solver

Let us now show that the Riemann solver derived in eqn. (3.42) is consistent. The third expression in eqn. (3.42) clearly shows that the flux produced by the Riemann solver is *consistent* in the sense that $\mathrm{F}^{(RS)} \to \mathrm{F}\left(\bar{\mathrm{U}}\right)$ as $\mathrm{U}_L \to \bar{\mathrm{U}}$ and $\mathrm{U}_R \to \bar{\mathrm{U}}$ for any state $\bar{\mathrm{U}}$. Thus for smooth mesh functions with vanishingly small jumps at the zone boundaries, the FDA in eqn. (3.38) approaches the PDE in eqn. (3.26) when the fluxes from the present Riemann solver are used in the FDA. A similar consistency will be demanded of the numerical fluxes that are used in the solution of non-linear hyperbolic systems.

Let us now turn our attention to the dissipation properties of the linear Riemann solver in eqn. (3.42). It is very important to relate what we are about to learn to the forward Euler and donor cell schemes from Chapter 2. We, therefore, urge the reader to review Sub-sections 2.7.1 and 2.7.5 from Chapter 2 before proceeding. Realize that both

the forward Euler and donor cell schemes are first order accurate in time. They differ only because the forward Euler scheme can be written in terms of fluxes that are second order accurate in space while the donor cell scheme uses fluxes that are only first order accurate in space. The expression in eqn. (3.42c) also shows us that the numerical flux from the Riemann solver can be decomposed into two parts. The first term, i.e., $\left(\mathrm{F}_R + \mathrm{F}_L\right)/2$, is a centered, second order flux. In the limit of scalar advection, such a centered, spatially second order accurate flux acting by itself would give us the forward Euler scheme. Just like the forward Euler scheme, the centered flux $\left(\mathrm{F}_R + \mathrm{F}_L\right)/2$, taken all by itself, would lead to numerical instability. The second term, i.e., $-\frac{1}{2}\left|\mathrm{A}\right|\left(\mathrm{U}_R - \mathrm{U}_L\right)$, should therefore be interpreted as the diffusive contribution to the Riemann solver; i.e., it is the part that suppresses the numerical instability. In the limit of scalar advection, the sum of the first and second terms gives rise to the donor cell flux. That flux is numerically stable. This decomposition of the flux from the Riemann solver into a centered part and a diffusive part is central to understanding the dissipation properties of any Riemann solver. We will put it to good use when we analyze the dissipation properties of various approximate Riemann solvers in a subsequent chapter.

When we consider schemes that are first order accurate in time and have no reconstructed sub-structure within each zone, we see that both the terms from eqn. (3.42c) are needed to achieve stability via upwinding. Schemes that use second order accurate TVD reconstruction will introduce a linear profile within each zone. That linear profile reduces the jump in the reconstructed variables at each zone boundary, resulting in reduced dissipation. Recall that the dissipation in eqn. (3.42c) is proportional to the jump $\left(\mathrm{U}_R - \mathrm{U}_L\right)$, with the result that reducing the jump at a zone boundary reduces the dissipation. Consequently, second order TVD schemes are substantially less dissipative than their donor cell-based cousins while retaining the advantage of monotonicity preserving propagation of flow features. However, it is important to keep in mind that second order upwind TVD schemes (as well as their higher order cousins, which we will

only study in subsequent chapters) also rely on stabilization via upwinding that is provided by the Riemann solver.

It is also important to observe that Eqns. (3.41), (3.42) and (3.43) have a structure that is based entirely on matrix manipulations. Roe (1981) showed that the solution of the Riemann problem for a non-linear hyperbolic system of conservation laws is closely approximated by a very similar matrix structure. Thus our investment in the study of the Riemann problem in this section will pay us a further dividend when we study the Riemann problem for non-linear systems of conservation laws. This completes our description of the Riemann problem for linear hyperbolic systems.

3.4.5) The Fluctuation Form

Fig. 3.10 shows that the Riemann problem evolves self-similarly. Any constant time slice of Fig. 3.10 with $t > 0$ looks just like any other constant time slice of Fig. 3.10 at a later time. In other words, as time evolves, the Riemann fan, and all the states associated with it, just spread out. But there is no fundamental change in the solution, other than this scaling. The Riemann problem, therefore, evolves self-similarly and this is an attribute that is also shared with the Riemann problem for non-linear hyperbolic systems of equations. Self-similar evolution means that the solution at any space-time point, (x,t) only depends on one variable – the similarity variable $\xi = x/t$. This is true for all $t > 0$. We can, therefore, write

$$\begin{aligned} \mathrm{U}(\xi) &= \mathrm{U}_L + \sum_{p=1}^{M} \alpha^p \, r^p \, \mathrm{H}(\xi - \lambda^p) \\ &= \mathrm{U}_R - \sum_{p=1}^{M} \alpha^p \, r^p \, \mathrm{H}(\lambda^p - \xi) \end{aligned} \qquad (3.aa)$$

Here $\mathrm{H}(x)$ is the Heaviside function; for $x \geq 0$, $\mathrm{H}(x) = 1$ whereas for $x < 0$, $\mathrm{H}(x) = 0$. Eqn. (3.aa) is identical to eqn. (3.37); it just shows more clearly how the contribution from each wave adds up to make the states in the Riemann fan. Eqn. (3.41) also shows how the each wave contributes to the flux. The individual contributions from each of the waves to the flux is referred to as a fluctuation. From eqn. (3.41a) we can see that the

numerical flux is made up of the left flux plus the fluctuations that are moving to the left. From eqn. (3.41b) we see that the numerical flux is also made up of the right flux minus the fluctuations that are moving to the right.

Using this notation of fluctuations, we can now write eqn. (3.38) in a more illustrative form. We can use eqn. (3.41a) two write $\bar{\mathrm{F}}^{n}_{i+1/2} = \mathrm{F}_i + \mathrm{A}^{-}\left(\mathrm{U}_{i+1} - \mathrm{U}_i\right)$ We can also use eqn. (3.41b) to write $\bar{\mathrm{F}}^{n}_{i-1/2} = \mathrm{F}_i - \mathrm{A}^{+}\left(\mathrm{U}_i - \mathrm{U}_{i-1}\right)$. Assembling the flux terms, we can write eqn. (3.38) in fluctuation form as

$$\bar{\mathrm{U}}^{n+1}_i = \bar{\mathrm{U}}^{n}_i - \frac{\Delta t}{\Delta x}\left(\mathrm{A}^{-}\left(\mathrm{U}_{i+1} - \mathrm{U}_i\right) + \mathrm{A}^{+}\left(\mathrm{U}_i - \mathrm{U}_{i-1}\right)\right) \qquad \text{(3.ab)}$$

The above equation illustrates that the solution $\bar{\mathrm{U}}^{n+1}_i$ at the later time is just the solution $\bar{\mathrm{U}}^{n}_i$ at the earlier time along with the contributions from the right-going fluctuations from the left zone boundary and the contributions from the left-going fluctuations from the right zone boundary. Notice that eqn. (3.ab) is not in conservation form. Even so, when the system is conservative, it retrieves the conservation form. For linear hyperbolic systems, this is not an important point of distinction because any linear hyperbolic system can always be recast in a conservation form. However, eqn. (3.ab) becomes important when considering non-linear hyperbolic systems that may not be in conservation form. We see that they can at least be written in a fluctuation form. The fluctuation form, therefore, establishes a desirable concordance between conservative systems and non-conservative hyperbolic systems.

Illustrating the Previous Section for the linearized Euler Equations

We now illustrate the ideas from this section for a representative system. The easiest way to obtain a physically meaningful linear hyperbolic system is to linearize a hyperbolic system of interest and then to freeze the coefficients in front of the space and time derivatives. Doing that for the Euler system after excluding the transverse velocities

and restricting it to one dimension gives us a linear hyperbolic system for the evolution of the density ρ , the x-velocity $\mathrm{v_x}$ and the pressure P which we catalogue below

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \mathrm{v_x}\\ \mathrm{P}\end{pmatrix} + \begin{pmatrix}\mathrm{v_{x0}} & \rho_0 & 0\\ 0 & \mathrm{v_{x0}} & \dfrac{1}{\rho_0}\\ 0 & \rho_0 \mathrm{c_0^2} & \mathrm{v_{x0}}\end{pmatrix} \frac{\partial}{\partial x}\begin{pmatrix}\rho\\ \mathrm{v_x}\\ \mathrm{P}\end{pmatrix} = 0$$

The above equation is equivalent to eqn. (1.58) from Section 1.5.2. Here ρ_0 , $\mathrm{v_{x0}}$ and $\mathrm{c_0}$ are the density, x-velocity and sound speed respectively around which we linearize the Euler system. This gives us a 3×3 linear hyperbolic system and the characteristic matrix is easily identified. The three ordered eigenvalues are given by eqn. (1.59). The orthonormalized right and left eigenvectors are given by eqns. (1.60) and (1.61) and they correspond to eqns. (3.27) and (3.28) respectively.

For any right and left states $\mathrm{U}_L = \left(\rho_L, \mathrm{v}_{xL}, \mathrm{P}_L\right)^T$ and $\mathrm{U}_R = \left(\rho_R, \mathrm{v}_{xR}, \mathrm{P}_R\right)^T$ respectively we can now formulate the Riemann problem for the linearized Euler system. We do this by projecting the jump in the variables into the space of the right eigenvectors as follows

$$\mathrm{U}_R - \mathrm{U}_L = \sum_{m=1}^{3} \alpha^m \, r^m \quad \text{where}$$

$$\alpha^1 \equiv \frac{-1}{2\,\mathrm{c}_0}\left(\mathrm{v}_{xR} - \mathrm{v}_{xL}\right) + \frac{1}{2\,\rho_0\,\mathrm{c}_0^2}\left(\mathrm{P}_R - \mathrm{P}_L\right) \;\; ; \;\; \alpha^2 \equiv \left(\rho_R - \rho_L\right) - \frac{1}{\mathrm{c}_0^2}\left(\mathrm{P}_R - \mathrm{P}_L\right) \;\; ;$$

$$\alpha^3 \equiv \frac{1}{2\,\mathrm{c}_0}\left(\mathrm{v}_{xR} - \mathrm{v}_{xL}\right) + \frac{1}{2\,\rho_0\,\mathrm{c}_0^2}\left(\mathrm{P}_R - \mathrm{P}_L\right)$$

The above equations correspond to eqns. (3.36). Assuming the initial discontinuity occurs at $x = 0$, its evolution in space and time is given by the similarity solution

$$
\begin{aligned}
\mathrm{U}(x,t) &= \mathrm{U}_L && \text{for } \frac{x}{t} < \mathrm{v}_{x0} - \mathrm{c}_0 \\
&= \mathrm{U}^{(1)} \equiv \mathrm{U}_L + \alpha^1 r^1 = \mathrm{U}_R - \alpha^2 r^2 - \alpha^3 r^3 && \text{for } \mathrm{v}_{x0} - \mathrm{c}_0 < \frac{x}{t} < \mathrm{v}_{x0} \\
&= \mathrm{U}^{(2)} \equiv \mathrm{U}_L + \alpha^1 r^1 + \alpha^2 r^2 = \mathrm{U}_R - \alpha^3 r^3 && \text{for } \mathrm{v}_{x0} < \frac{x}{t} < \mathrm{v}_{x0} + \mathrm{c}_0 \\
&= \mathrm{U}_R && \text{for } \mathrm{v}_{x0} + \mathrm{c}_0 < \frac{x}{t}
\end{aligned}
$$

The above equation corresponds to eqn. (3.37).

Let us take the transonic case with a positive fluid velocity, so that $0 < \mathrm{v}_{x0} < \mathrm{c}_0$. We then have m_0=1 because $\mathrm{v}_{x0} - \mathrm{c}_0 < 0$, $\mathrm{v}_{x0} > 0$ and $\mathrm{v}_{x0} + \mathrm{c}_0 > 0$. We can then set

$$
\begin{aligned}
\mathrm{F}^{(RS)} = \mathrm{A}\,\mathrm{U}^{(1)} &= \mathrm{A}\,\mathrm{U}_L + \lambda^1 \alpha^1 r^1 \\
&= \mathrm{A}\,\mathrm{U}_R - \lambda^2 \alpha^2 r^2 - \lambda^3 \alpha^3 r^3 \\
&= \frac{1}{2}\left(\mathrm{A}\,\mathrm{U}_R + \mathrm{A}\,\mathrm{U}_L\right) - \frac{1}{2}\left(\left|\lambda^1\right| \alpha^1 r^1 + \left|\lambda^2\right| \alpha^2 r^2 + \left|\lambda^3\right| \alpha^3 r^3\right)
\end{aligned}
$$

The above equation gives us the resolved flux for the Riemann problem and corresponds to eqn. (3.41). Fig. 3.11 also illustrates the transonic case for a system with three wave families. For the present transonic problem we can now define

$$
\Lambda^+ \equiv \text{diag}\left\{0,\ \mathrm{v}_{x0},\ \mathrm{v}_{x0} + \mathrm{c}_0\right\} \quad ; \quad \Lambda^- \equiv \text{diag}\left\{\mathrm{v}_{x0} - \mathrm{c}_0,\ 0,\ 0\right\} \quad ;
$$
$$
\left|\Lambda\right| \equiv \left\{\mathrm{c}_0 - \mathrm{v}_{x0},\ \mathrm{v}_{x0},\ \mathrm{v}_{x0} + \mathrm{c}_0\right\}
$$

Notice that all the elements in Λ^+ are non-negative, all the elements in Λ^- are non-positive and that $\left|\Lambda\right| = \Lambda^+ - \Lambda^-$. With these matrices in hand, we can proceed to build A^+ , A^- and $\left|\mathrm{A}\right|$ using eqn. (3.43) and the eigenvectors that we have developed in this box. Eqn. (3.42) then enables us to solve any Riemann problem associated with this linear hyperbolic system. As a result, the fluxes needed in eqn. (3.38) can always be obtained.

3.5) Numerical Boundary Conditions for Linear Hyperbolic Systems

To compute the solution of any scientific or engineering problem involving hyperbolic PDEs we have to solve the problem on a specified computational domain. The solution that we eventually get depends on the physics of the PDE, the initial conditions and the information that comes in from the boundaries as the problem is evolved in time. This is known as the *initial boundary value problem*. Notice, therefore, that the specification of values at the boundary can play an important role in the evolution of the problem. It is not always easy to specify the boundary conditions as they are often based on making an intelligent prognostication of the solution outside the boundary. Yet we assume that the interesting part of the computational problem and its evolution is captured within the computational domain, so that the boundary conditions have only a gentle and predictable influence on the interior of the domain. Even so, the boundary conditions cannot be specified entirely arbitrarily. For example, for the scalar advection problem shown in Fig. 2.15, the advection takes place to the right. If we were to solve that problem on a finite, one-dimensional, non-periodic computational domain, the characteristics would bring in new information from the left boundary. Thus the value of the desired solution should be specified for all times at the left boundary. However, specifying the solution at the right boundary would indeed over-specify the problem. The right boundary should be such as to permit any solution that approaches it to smoothly leave the domain without injecting any spurious information back into the domain.

For an "*M*" component linear hyperbolic system eqn. (3.29) has shown us that the m^{th} eigenweight propagates with a speed λ^m . As in the previous section we assume an ordered set of eigenvalues. Because the hyperbolic system is linear and the characteristic matrix is a constant, the wave speeds do not change as the solution evolves. As in eqn. (3.39) or Fig. 3.10, we assume that there is some integer m_0 such that $\lambda^{m_0} < 0 \le \lambda^{m_0+1}$. Say that the linear hyperbolic system is to be evolved on a one-dimensional, non-periodic computational domain. Then, at the left boundary, we want the waves associated with outgoing characteristics to flow out smoothly so that we would want the first m_0 waves

(with their associated eigenweights) to leave the domain without generating any back-reaction. The characteristics for the next $M - m_0$ waves enter the computational domain from the left so that we should specify the next $M - m_0$ eigenweights at the left boundary. At the right boundary we also want the waves associated with the outgoing characteristics to leave the domain smoothly so that we would want the last $M - m_0$ waves (with their associated eigenweights) to leave the domain without generating any back-reaction. The characteristics for the first m_0 waves enter the computational domain from the right so that we should specify the first m_0 eigenweights at the right boundary. We, therefore, see that for this very simple linear hyperbolic problem a total of exactly "*M*" eigenweights must be specified at the boundaries and allowance must be made for the same number of eigenweights to leave the domain without generating a back-reaction. Boundary conditions that permit a wave to leave the computational domain without generating a back-reaction are called *radiative or non-reflective boundary conditions*. Such non-reflective boundary conditions were first designed by Hedstrom (1979). Boundary conditions that specify the amplitude of a wave that should flow into a computational domain are called *inflow boundary conditions*. The above-mentioned two types of boundary condition are but a very small subset of the kinds of boundary conditions that are used in practice.

Certain types of boundary conditions are indeed specific to the physics represented by a particular type of hyperbolic system. As a result, it is not possible to give a complete catalogue of boundary conditions. For the non-linear Euler and Navier-Stokes equations a systematic effort has been made to catalogue all the different types of boundary conditions, see Thompson (1990), Sutherland & Kennedy (2003) and Liu & Vasilyev (2010), but even such an effort is intimately tied to the type of scheme being used. For certain steady state aerodynamics problems, the far field solutions are known to a good approximation (Jameson 1982, Yee *et al.* 1982). Unfortunately, this advantage does not extend to problems in astrophysics and space science. It is, however, possible to improve one's intuition so that one can always develop suitable boundary conditions as the need arises.

Fig. 3.12 shows a schematic representation of a mesh spanning the domain $x \in [a,b]$ with "*N*" zones and $\Delta x = (b-a)/N$. The zone centers and zone boundaries are given by $x_i = a + (i - 1/2)\,\Delta x$ and $x_{i+1/2} = a + i\,\Delta x$ respectively with integral values for "*i*". The mesh function in zone "*i*" at a time t^n is given by $\bar{U}_i^n$. Our philosophy in developing the boundary conditions is that we should apply the same numerical algorithm, if this is at all possible, to all the zones of the mesh. Notice that our numerical algorithms consist of applying limiters to obtain undivided differences within each zone, so that we will need one more zone on each side of any zone to which the limiter is applied. We will, therefore, need a few *ghost zones* on either side of the dynamically active zones in the mesh, see Fig. 3.12. These ghost zones will be filled in with appropriate data so as to make the solution that is interior to the mesh behave appropriately. To apply the Riemann solvers to zone boundaries $x_{1/2}$ and $x_{N+1/2}$ we need solution values as well as undivided differences to be specified in ghost zones "0" and "*N*+1". As a result, when designing a second order accurate TVD scheme we will at least need to specify the solution in ghost zones "–1" and "*N*+2". In other words, we need a minimum of two ghost zones on either side of our computational domain.

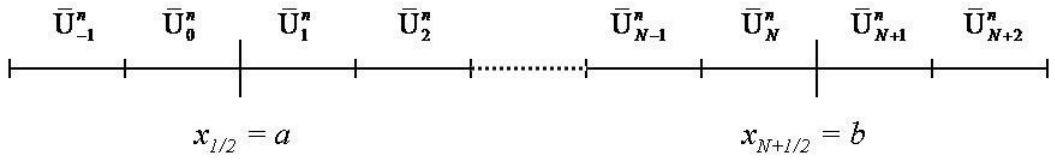


Fig. 3.12 shows a one-dimensional mesh with "N" zones covering [a,b]. The values of the zone-centered mesh function are shown at time t^n . The ghost zones, at which the boundary conditions are specified, are also shown.

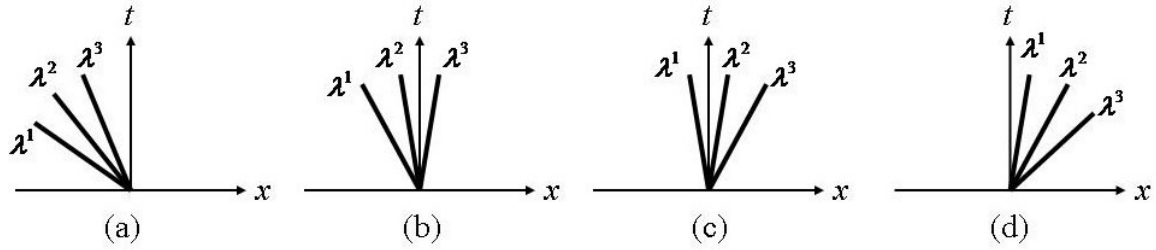
Let us now focus on the left boundary $x = a$. Eqn. (3.29) tells us that the m^{th} eigenweight, w^m, evolves according to the equation $w_t^m + \lambda^m\, w_x^m = 0$. Thus to suppress any back-reaction from the first m_0 waves at the boundary $x = a$ we need to make $w_x^m = 0$ for $m \le m_0$ which will, in turn, make $w_t^m = 0$ for those same wave families. In

other words, a spatially varying eigenweight results in a temporally propagating wave of a particular family. We wish to suppress w_t^m at the left boundary for outgoing waves, i.e. all of the m^{th} waves with $m \le m_0$. This is easily achieved by zeroing the spatial variation in those modes, i.e. by setting $w_{-1}^m = w_0^m = w_1^m$ for $m \le m_0$ at each timestep in the solution process. Here w_{-1}^m, w_0^m and w_1^m are the eigenweights of the m^{th} wave family in the zones "–1", "0" and "1" respectively. Thus w_1^m is used to refresh the values w_0^m and w_{-1}^m at each timestep. Notice that this corresponds to a zeroth order accurate extrapolation. Such extrapolations have been found to be numerically stable; higher order extrapolations tend to be numerically unstable. This completes our description of the non-reflective boundary conditions at the left boundary.

Now consider the wave families with $m > m_0$ at the same left boundary, $x = a$. Theses waves carry information into the computational domain. For these waves, we can actually set w_{-1}^m and w_0^m in any time-evolving fashion that is consistent with the waves we want to propagate into the left boundary $x = a$. For example, say we wish to make a space and time dependent solution propagate into the computational domain from the left along a characteristic with $m > m_0$. We can do that by endowing w_{-1}^m and w_0^m with that space and time-dependence and it will be propagated into the computational domain; see the box at the end of this section for an example. In doing so, one has to be mindful of the intrinsic numerical resolution of the scheme so that the spatial variation in the ghost zones' values should be distributed over several zones and substantial temporal variation in those values should only occur over several timesteps. This completes our description of the inflow boundary conditions at the left boundary. Similar considerations can be made for the right boundary. Note, therefore, that all the eigenweights are specified in all the ghost zones making it possible for us to retrieve the solution vector "U" in each of the four ghost zones of Fig. 3.12. To take zone "–1" as an example, we obtain the solution vector in that zone as $\bar{\mathrm{U}}_{-1}^n = \sum_{m=1}^{M} w_{-1}^m \, r^m$.

Please observe that our description of boundary conditions is not comprehensive. For example, *periodic boundary conditions* are most easily enforced by setting $\bar{U}_0^n = \bar{U}_N^n$, $\bar{U}_{-1}^n = \bar{U}_{N-1}^n$, $\bar{U}_{N+1}^n = \bar{U}_1^n$ and $\bar{U}_{N+2}^n = \bar{U}_2^n$. We might have situations where we wish to have zero gradients at a boundary in order to ensure that a feature in the solution that is approaching a boundary can leave the boundary without producing any new information in the computational domain. For the class of higher order accurate Godunov schemes that we explore here, this is most easily achieved by *outflow or continuitive boundary conditions* which continuously extend the solution from the last interior zone into the ghost zones. Such outflow boundary conditions are most easily enforced at the left boundary by setting $\bar{U}_{-1}^n = \bar{U}_0^n = \bar{U}_1^n$ and at the right boundary by setting $\bar{U}_{N+2}^n = \bar{U}_{N+1}^n = \bar{U}_N^n$. Outflow boundary conditions work very nicely when Riemann solvers are used at domain boundaries to evaluate the flux. This is because the Riemann solver relies on examining the wave structure of the problem and, therefore, permits the outgoing solution features to leave the domain in a natural way. The box below gives a description of a few other boundary conditions that pertain to the linearized Euler equations.

Boundary Conditions for the Linearized Euler Equations



The four figures above show the different ways in which the characteristic propagate in space-time for the linearized Euler equations. The characteristics are shown as thick lines. The propagation speeds of the characteristics depend on the relative magnitudes of v_{x0} *and* c_0.

In this box we consider the linearized Euler equations that were first documented in the box at the end of Section 3.4. It is a linear hyperbolic system with three

eigenvalues. For a given choice of v_{x0} and c_0 , the evolution of the characteristics in space-time is set once and for all and is shown by the four diagrams above. The ratio $|v_{x0}|/c_0$ is called the *Mach number* of the flow. The Mach number plays an important role in determining the nature of the solution as well as the form of the boundary conditions. In this box we assume that we utilize the same mesh as in Fig. 3.12. Thus $v_{x0} < -c_0$ corresponds to the supersonic situation shown in figure (a) of this box where we take $m_0 = 3$ and consider all three waves to be outgoing at the left boundary and incoming at the right boundary. Similarly, $|v_{x0}| \le |c_0|$ with $v_{x0} < 0$ corresponds to the subsonic situation shown in figure (b) where we take $m_0 = 2$. Thus the first two waves are outgoing at the left boundary and incoming at the left boundary, while the third wave is outgoing at the right boundary and incoming at the left boundary. Likewise, $|v_{x0}| \le |c_0|$ with $v_{x0} \ge 0$ corresponds to figure (c) where we take $m_0 = 1$. Consequently, the first wave is outgoing at the left boundary and incoming at the right boundary, while the second and third waves are outgoing at the right boundary and incoming at the left boundary. Finally, $v_{x0} > c_0$ corresponds to the supersonic situation shown in figure (d) of this box where we take $m_0 = 0$ so that all waves are incoming at the left boundary and outgoing at the right boundary.

In many CFD problems we want flow features to reflect back from a solid surface. If we want to set up *reflective boundary conditions* at the right boundary then this can be achieved by setting

$$\begin{pmatrix} \rho \\ v_x \\ P \end{pmatrix}_{N+1}^{n} = \begin{pmatrix} \rho \\ -v_x \\ P \end{pmatrix}_{N}^{n} \quad \text{and} \quad \begin{pmatrix} \rho \\ v_x \\ P \end{pmatrix}_{N+2}^{n} = \begin{pmatrix} \rho \\ -v_x \\ P \end{pmatrix}_{N-1}^{n}$$

The two ghost zones that lie rightward of the right boundary have to be reset at each timestep according to the above rules. Any wave approaching the right boundary will then be reflected back into the domain. Notice that this boundary condition is specific to

the physics of the linear hyperbolic system that we consider here. Other hyperbolic problems will have other boundary conditions that are specific to the science that they represent.

Let us say that we have $|v_{x0}| \le |c_0|$ with $v_{x0} < 0$, which corresponds to figure (b) above. We may then want to set up an *oscillatory boundary condition* at the left boundary as follows. Say we want to send in a sinusoidally oscillating solution from the left boundary along the third (incoming) wave without generating any new information in the outgoing waves at that boundary. We can arrange for oscillations with a wave number "*k*" and amplitude A_0 to come in through the left boundary on the third characteristic by setting

$$\bar{U}_0^n = w_1^1 r^1 + w_1^2 r^2 + A_0 \sin\left[k\left(a - \Delta x/2 - \lambda^3 t^n\right)\right] r^3$$
$$\bar{U}_{-1}^n = w_1^1 r^1 + w_1^2 r^2 + A_0 \sin\left[k\left(a - 3\Delta x/2 - \lambda^3 t^n\right)\right] r^3$$

Notice that the first two characteristic fields have been extrapolated from the first interior zone, i.e. we have set $w_{-1}^1 = w_0^1 = w_1^1$ and $w_{-1}^2 = w_0^2 = w_1^2$ in the above two equations. This extrapolation has to be carried out anew at each time step. The last characteristic field carries the ingoing wave into the computational domain. For a typical second order scheme, the wavelength that corresponds to "*k*" should be larger than ten to twenty zones in order for the wave to propagate without significant attenuation on the mesh.

3.6) Second Order Upwind Schemes for Linear Hyperbolic Systems

Having understood the nature of linear hyperbolic systems in the previous two sections, we now wish to apply our new-found knowledge to their numerical solution. Eqn. (3.29) has shown us that the numerical solution of the linear *M*×*M* hyperbolic system $U_t + A U_x = 0$ is most easily obtained by examining the "*M*" scalar advection

equations $w_t^m + \lambda^m w_x^m = 0$ where $w^m \equiv l^m \cdot U$. Recall that λ^m, r^m, l^m and w^m are the eigenvalue, right eigenvector, left eigenvector and eigenweight respectively of the m^{th} family of simple wave. Since the methods developed in Sections 3.2 and 3.3 are very adept at scalar advection, we realize that we can build solution strategies for linear hyperbolic systems that propagate each of the "*M*" characteristic weights at their appropriate speeds. Indeed we present three such numerical schemes.

Sub-section 3.6.1 presents a direct upgrade of the Lax-Wendroff method, this time modified by the application of limiters. Such a TVD method was first presented by Harten (1983). It has the advantage that it is fast because the entire second order accurate time-update is accomplished in one step.

The second method uses a two-stage, second order accurate Runge-Kutta time stepping strategy and is presented in Sub-section 3.6.2. Each stage of a Runge-Kutta method is simpler, making it easier to add on additional physics. This makes Runge-Kutta methods very useful for many science and engineering problems. However, the scheme consists of two stages and is, therefore, costlier. Such Runge-Kutta timestepping strategies can be extended to higher orders (Shu and Osher 1988, Shu 1988) and Chapter 7 will show how this is done. In contrast, the Lax-Wendroff scheme is usually restricted to being second order accurate in time because it becomes progressively harder to trade time derivatives for spatial derivatives for an entire scheme as the order of accuracy is increased. Note though that the Lax-Wendroff procedure from Sub-section 2.7.3 is easier to extend to higher orders.

Sub-section 3.6.3 presents another scheme by Colella (1985) where the predictor step is taken within each zone by using the slope information within that zone in the most economical fashion. Such schemes will also be extended to higher order in Chapter 7 using the ADER methodology (Titarev and Toro 2002, Dumbser *et al.* 2008). ADER is an acronym that stands for Arbitrary accuracy DERivative Riemann problem.

The schemes in Sub-sections 3.6.1 to 3.6.3 present the three most important design strategies for second order accurate schemes. Sub-section 3.6.4 shows numerical results from these three schemes.

We describe all the schemes on a one-dimensional mesh in the *x*-direction having a uniform zone size Δx. Fig. 3.12 shows a schematic representation of such a mesh spanning the domain $x \in [a,b]$ with "*N*" zones and $\Delta x = (b-a)/N$. The zone centers and zone boundaries are given by $x_i = a + (i - 1/2)\,\Delta x$ and $x_{i+1/2} = a + i\,\Delta x$ respectively with integral values for "*i*". We describe a single time-step where the solution vector is taken from a time t^n to a time $t^{n+1} = t^n + \Delta t$. The full time-evolution of the hyperbolic system is obtained on a computer by repeating the timestep as many times as is needed. Because the hyperbolic system is linear we make the simplifying assumption that the eigenvalues and the right and left eigenvectors are fully specified by evaluating them only once in the numerical implementation. We use a zone centered collocation of data so that at time t^n the zone averaged vector of "*M*" conserved variables in the zone "*i*" is given by $\bar{U}_i^n$. Each of the methods describes the procedure for obtaining $\bar{U}_i^{n+1}$ for all zones "*i*" at time t^{n+1}. All of the methods presented here are based on the non-linear hybridization ideas that were developed in Sections 3.2 and 3.3. Consequently, the present schemes are stable with a CFL number of unity in one dimension. Since our present goal is not to describe a production-grade scheme, we assume that the solution is suitably initialized with constant values in two extra ghost zones that lie on either side of the mesh being considered. Our present examples will be easy enough to get by with this simplification.

3.6.1) Second Order Accurate Extension of the Lax-Wendroff Scheme with Limiters

Since the linear hyperbolic system displays its most natural form in the characteristic variables, we construct characteristic variables within each zone "*i*" as follows

$$w_i^m = l^m \ \bar{\mathrm{U}}_i^n \quad \text{for } m = 1,...,M \tag{3.44}$$

Notice that w_i^m has to be constructed anew at each timestep using $\bar{\mathrm{U}}_i^n$ that is available at the beginning of each timestep. (We have not included the superscript "n" in our definition of w_i^m just to keep the notation uncluttered.) Eqn. (3.29) then tells us that we can think of eqn. (3.44) as providing us with "*M*" scalar mesh functions $\left\{w_i^m\right\}$ each of which has to be advected with a speed λ^m. Section 3.2 has already shown us how to carry out such an advection with second order accuracy. Thus our next step is to build limited slopes for these characteristic variables. We build limited slopes for the characteristic variables within each zone "*i*" as follows

$$\Delta w_i^m = Limiter\left(w_{i+1}^m - w_i^m, w_i^m - w_{i-1}^m\right) \quad \text{for } m = 1,...,M \tag{3.45}$$

Here "*Limiter*" stands for any of the slope limiters catalogued in the box at the end of Section 3.2.

Notice that the m^{th} characteristic variable in zone "*i*" , i.e. w_i^m, is advected as a scalar with a speed λ^m using scalar fluxes $\mathrm{f}_{i+1/2}^m$ and $\mathrm{f}_{i-1/2}^m$ for that characteristic field. The scalar fluxes $\mathrm{f}_{i+1/2}^m$ and $\mathrm{f}_{i-1/2}^m$ are defined at zone boundaries "*i*+1/2" and "*i*–1/2" respectively. Thus an important ingredient of the present scheme consists of specifying $\mathrm{f}_{i+1/2}^m$. The tricky part in the specification of the flux $\mathrm{f}_{i+1/2}^m$ consists of realizing that λ^m can be positive or negative. If λ^m were just positive, eqn. (3.17) would have served us very well. As it stands, we need an upgraded expression that is based on eqn. (3.17) but can work for both signs of λ^m. When $\lambda^m \geq 0$, such a flux $\mathrm{f}_{i+1/2}^m$ at zone boundary "*i*+1/2" must depend on the upwind variables w_i^m and Δw_i^m. When $\lambda^m < 0$, the flux $\mathrm{f}_{i+1/2}^m$ at zone boundary "*i*+1/2" must depend on the upwind variables w_{i+1}^m and Δw_{i+1}^m. Following a convention that is customary in the definition of this scheme, we define

$$\begin{aligned} \Delta w_{i+1/2}^{m} &= \Delta w_{i}^{m} \qquad \text{for } \lambda^{m} \geq 0 \\ &= \Delta w_{i+1}^{m} \qquad \text{for } \lambda^{m} < 0 \end{aligned} \tag{3.46}$$

We can then write $\mathrm{f}_{i+1/2}^{m}$ at each zone boundary "i+1/2" as

$$\begin{aligned} \mathrm{f}_{i+1/2}^{m} &= \lambda^{+,m}\left[w_{i}^{m} + \frac{1}{2}\left(1 - \frac{\Delta t}{\Delta x}\left|\lambda^{m}\right|\right)\Delta w_{i+1/2}^{m} \right] \\ &\quad + \lambda^{-,m}\left[w_{i+1}^{m} - \frac{1}{2}\left(1 - \frac{\Delta t}{\Delta x}\left|\lambda^{m}\right|\right)\Delta w_{i+1/2}^{m} \right] \qquad \text{for } m = 1,\ldots,M \end{aligned} \tag{3.47}$$

Recall that only one of the two terms $\lambda^{+,m}$ or $\lambda^{-,m}$ is non-zero for the m^{th} wave family. Notice that eqn. (3.47) is still not the entire flux vector that we seek. Just as the entire solution vector in zone "i" can be written as $\bar{\mathrm{U}}_{i}^{n} = \sum_{m=1}^{M} w_{i}^{m}\, r^{m}$ (i.e. we have to sum the right eigenvectors with the eigenweights), we can write the entire flux vector at zone boundary "i+1/2" as $\bar{\mathrm{F}}_{i+1/2}^{n+1/2} = \sum_{m=1}^{M} \mathrm{f}_{i+1/2}^{m}\, r^{m}$. Notice too that eqn. (3.47) is second order accurate in space and time. Hence the resulting flux $\bar{\mathrm{F}}_{i+1/2}^{n+1/2}$ is written with a superscript "n+1/2" to show that it is second order accurate in space and time and, therefore, properly time-centered. Using the identities in eqn. (3.43) then permits us to write

$$\bar{\mathrm{F}}_{i+1/2}^{n+1/2} = \mathrm{A}^{+}\,\bar{\mathrm{U}}_{i}^{n} + \mathrm{A}^{-}\,\bar{\mathrm{U}}_{i+1}^{n} + \tilde{\mathrm{F}}_{i+1/2} \text{ where } \tilde{\mathrm{F}}_{i+1/2} = \frac{1}{2}\sum_{m=1}^{M}\left|\lambda^{m}\right|\left(1 - \frac{\Delta t}{\Delta x}\left|\lambda^{m}\right|\right)\Delta w_{i+1/2}^{m}\, r^{m} \tag{3.48}$$

Notice that the flux $\bar{\mathrm{F}}_{i+1/2}^{n+1/2}$ in eqn. (3.48) lends itself to an easy interpretation. It is made up of three parts that are each easy to interpret. The part $\mathrm{A}^{+}\,\bar{\mathrm{U}}_{i}^{n}$ carries the contributions from all the right-going waves with $\lambda^{m} \geq 0$. It represents a flux that is first order accurate in space and time from just those right-going waves. The part $\mathrm{A}^{-}\,\bar{\mathrm{U}}_{i+1}^{n}$ carries the contributions from all the left-going waves with $\lambda^{m} < 0$. This part also represents a flux that is first order accurate in space and time from just those left-going waves. As

with eqns. (3.16) and (3.17), the flux $\overline{\mathrm{F}}_{i+1/2}^{n+1/2}$ in eqn. (3.48) only becomes second order accurate in space and time if we include an anti-diffusive part, i.e. the third part $\tilde{\mathrm{F}}_{i+1/2}$ in eqn. (3.48). All the terms in eqn. (3.48) are fully specified by our eqns. (3.44), (3.45), (3.46) and (3.43). By building the fluxes $\overline{\mathrm{F}}_{i+1/2}^{n+1/2}$ at all the zone boundaries "i+1/2" we obtain the final, one-step, second order accurate update

$$\overline{\mathrm{U}}_i^{n+1} = \overline{\mathrm{U}}_i^n - \frac{\Delta t}{\Delta x}\left(\overline{\mathrm{F}}_{i+1/2}^{n+1/2} - \overline{\mathrm{F}}_{i-1/2}^{n+1/2}\right) \tag{3.49}$$

This completes our description of the original TVD scheme of Harten (1983) for linear hyperbolic systems.

3.6.2) Second Order Accurate, Two-Stage Runge-Kutta Scheme with Limiters

The second order accurate, two-stage Runge-Kutta scheme is based on a simple philosophy. The time-evolution in such schemes is based on time-explicit, second order accurate Runge-Kutta methods for solving ordinary differential equations. I.e., we write the hyperbolic system as $\mathrm{U}_t = -\mathrm{F}_x$ and treat the gradient of the fluxes as if it is the right hand side of a system of ordinary differential equations that are to be evolved in time. Such Runge-Kutta methods are based on the idea of having a sequence of stages, each of which is simple and may only be first order accurate in time. The gradient of the fluxes will indeed have to be evaluated at each of those stages. However, when the result of all the stages is assembled together, the resulting Runge-Kutta scheme is higher order accurate in time. Consequently, each of the stages can be simpler than a single stage scheme that seeks to achieve second or higher order accuracy in time. An example of the latter would be the Lax-Wendroff scheme from the previous sub-section. When this philosophy is applied to the numerical solution of hyperbolic PDEs, the goal is to build the fluxes at each stage using only the solution at that stage. Thus the fluxes only have to be second order accurate in space, though not necessarily second order accurate in time. The whole scheme is, however, second order accurate in time. The boundary conditions

have to be consistently applied at each of the two stages. The approach described here functionally splits off the task of constructing a higher order accurate spatial flux from the task of ensuring higher order accurate temporal evolution of the hyperbolic PDE. The temporal accuracy is always matched to the spatial accuracy. It is also known in the literature as a *method of lines approach* or a *semi-discrete* method because the time-evolution is enforced by some suitable ODE method that has a sequence of discrete stages.

The stages in the second order accurate, two-stage Runge-Kutta scheme are given by the following two equations

$$\begin{aligned} \bar{\mathrm{U}}_i^{n+1/2} &= \bar{\mathrm{U}}_i^n - \frac{\Delta t}{2\Delta x}\left(\bar{\mathrm{F}}_{i+1/2}^n\left(\bar{\mathrm{U}}^n\right) - \bar{\mathrm{F}}_{i-1/2}^n\left(\bar{\mathrm{U}}^n\right)\right) \\ \bar{\mathrm{U}}_i^{n+1} &= \bar{\mathrm{U}}_i^n - \frac{\Delta t}{\Delta x}\left(\bar{\mathrm{F}}_{i+1/2}^{n+1/2}\left(\bar{\mathrm{U}}^{n+1/2}\right) - \bar{\mathrm{F}}_{i-1/2}^{n+1/2}\left(\bar{\mathrm{U}}^{n+1/2}\right)\right) \end{aligned} \qquad (3.50).$$

The above scheme is also called the *modified Euler approximation*. The first line in eqn. (3.50) is called the predictor step of the Runge-Kutta scheme. In this step we build the fluxes $\bar{\mathrm{F}}_{i+1/2}^n\left(\bar{\mathrm{U}}^n\right)$ so that they are second order accurate in space but only first order accurate in time using the mesh function $\left\{\bar{\mathrm{U}}^n\right\}$ at time t^n. (We will provide all the details associated with obtaining a second order flux from the mesh function in a couple of paragraphs.) Consequently, the mesh function $\left\{\bar{\mathrm{U}}^{n+1/2}\right\}$ that we obtain after applying the first stage to the entire mesh is only first order accurate in time and can be thought of as corresponding to an intermediate time $t^{n+1/2} = t^n + \Delta t/2$. Technically, it is referred to as an *internal stage* of the Runge-Kutta scheme. When implementing the scheme on a computer, notice that we have to store both the mesh functions $\left\{\bar{\mathrm{U}}^n\right\}$ and $\left\{\bar{\mathrm{U}}^{n+1/2}\right\}$ because they are both needed in the second line of eqn. (3.50). Using the newly obtained $\left\{\bar{\mathrm{U}}^{n+1/2}\right\}$ we can obtain the fluxes $\bar{\mathrm{F}}_{i+1/2}^{n+1/2}\left(\bar{\mathrm{U}}^{n+1/2}\right)$. Because these fluxes are properly time-centered and also second order accurate in space, they can provide an update in eqn. (3.50) that is second order accurate in space and time. The second stage of eqn. (3.50) is

often called the corrector step of the Runge-Kutta scheme and yields a second order accurate mesh function $\left\{\overline{\mathrm{U}}^{n+1}\right\}$ at the time $t^{n+1}=t^n+\Delta t$. Because of its easy interpretation, eqn. (3.50) is well-used in practice.

The *improved Euler approximation* is also has some noteworthy features. It is a two-stage Runge-Kutta scheme given by

$$\begin{aligned}\overline{\mathrm{U}}_i^{(1)} &= \overline{\mathrm{U}}_i^n - \frac{\Delta t}{\Delta x}\left(\overline{\mathrm{F}}_{i+1/2}^n\left(\overline{\mathrm{U}}^n\right) - \overline{\mathrm{F}}_{i-1/2}^n\left(\overline{\mathrm{U}}^n\right)\right)\\ \overline{\mathrm{U}}_i^{n+1} &= \frac{1}{2}\left(\overline{\mathrm{U}}_i^n + \overline{\mathrm{U}}_i^{(1)}\right) - \frac{\Delta t}{2\Delta x}\left(\overline{\mathrm{F}}_{i+1/2}^{(1)}\left(\overline{\mathrm{U}}^{(1)}\right) - \overline{\mathrm{F}}_{i-1/2}^{(1)}\left(\overline{\mathrm{U}}^{(1)}\right)\right)\end{aligned} \tag{3.51}$$

Here the mesh function $\left\{\overline{\mathrm{U}}_i^{(1)}\right\}$ is an internal stage of the scheme. It is worth mentioning that Shu and Osher (1988) and Shu (1988) showed that the above scheme has a special property that if each of the stages in the scheme is TVD then the whole scheme will be TVD. Runge-Kutta time stepping schemes that have such a property are referred to as *strong stability preserving* (SSP) time discretizations, see Spiteri and Ruuth (2003). In practice, eqns. (3.50) and (3.51) perform equally well, even though eqn. (3.50) is not SSP. The SSP property for eqn. (3.51) is explored further in the box at the end of this sub-section where it is shown that it is certainly SSP for scalar advection and provisionally for linear hyperbolic systems.

To provide a complete description of the scheme we only need to describe the procedure for obtaining $\overline{\mathrm{F}}_{i+1/2}\left(\overline{\mathrm{U}}\right)$ from a mesh function $\left\{\overline{\mathrm{U}}\right\}$. This is because the procedure for building the fluxes from a given mesh function is indeed identical in each of the two stages of the Runge-Kutta schemes. The steps that are carried out in any one of the stages are shown schematically in Fig. 3.13. As in the previous Sub-section, we realize that since the characteristic variables are indeed the entities that are advected, it is most natural to carry out the reconstruction by limiting the characteristic variables. This is done in the following three steps. First, we use the left eigenvectors to project the solution on to the characteristic variables

$$w_i^m = l^m \ \bar{\mathrm{U}}_i \quad \text{for } m = 1,...,M \tag{3.52}$$

Second, we limit the characteristic variables

$$\Delta w_i^m = Limiter\left(w_{i+1}^m - w_i^m, w_i^m - w_{i-1}^m\right) \quad \text{for } m = 1,...,M \tag{3.53}$$

Third, we use the limited slopes of the characteristic variables to obtain the limited slopes of the conserved variables. This is done by using the right eigenvectors as follows

$$\overline{\Delta \mathrm{U}}_i = \sum_{m=1}^{M} \Delta w_i^m \ r^m \tag{3.54}$$

Taken together, eqns. (3.52), (3.53) and (3.54) constitute a procedure for obtaining a properly limited vector of slopes $\overline{\Delta \mathrm{U}}_i$ within each zone. This method for reconstructing the solution is called *characteristic reconstruction* or *limiting on the characteristic variables*. While the projection to the space of characteristic variables in eqn. (3.52) and the transcription back to physical variables in eqn. (3.54) might be computationally expensive, characteristic reconstruction is truest to the inner structure of the linear hyperbolic system which tells us that the characteristic variables undergo scalar advection with their characteristic speeds.

Once the reconstructed vector of slopes $\overline{\Delta \mathrm{U}}_i$ is available within each zone, as shown by the dashed lines in the upper panel of Fig. 3.13, it is possible to specify the solution at any point on the mesh with second order accuracy in space. In particular, we can specify it at the zone boundary "i+1/2". Notice that we have two possible values at the zone boundary "i+1/2" given by

$$\mathrm{U}_{L;i+1/2} \equiv \bar{\mathrm{U}}_i + \frac{1}{2} \overline{\Delta \mathrm{U}}_i \quad ; \quad \mathrm{U}_{R;i+1/2} \equiv \bar{\mathrm{U}}_{i+1} - \frac{1}{2} \overline{\Delta \mathrm{U}}_{i+1} \tag{3.55}$$

$\mathrm{U}_{L;i+1/2}$ is defined immediately to the left of the zone boundary "i+1/2" using values from zone "i". $\mathrm{U}_{R;i+1/2}$ is defined immediately to the right of the zone boundary "i+1/2" using values from zone "i+1". Please see the upper panel of Fig. 1.13. Our development of the Riemann solver in Sub-sections 3.4.2 and 3.4.3 now shows its utility. The flux $\overline{\mathrm{F}}_{i+1/2}$ that we seek at the zone boundary "i+1/2" is nothing but the resolved flux from the Riemann problem with $\mathrm{U}_{L;i+1/2}$ as the left state and $\mathrm{U}_{R;i+1/2}$ as the right state. In other words, we can think of the Riemann solver, F_{RS} , as a machine that takes two states and produces one resolved flux that is to be used in the numerical scheme. This is shown in the lower panel of Fig. 1.13. Thus at all zone boundaries "i+1/2" we invoke the machinery of the Riemann solver to obtain

$$\overline{\mathrm{F}}_{i+1/2} = \mathrm{F}_{RS}\left(\mathrm{U}_{L;i+1/2},\ \mathrm{U}_{R;i+1/2}\right) \tag{3.56}$$

This flux can now be used in eqns. (3.50) or (3.51) depending on which stage of the Runge-Kutta scheme we are evaluating. This completes our description of the two-stage Runge-Kutta scheme.

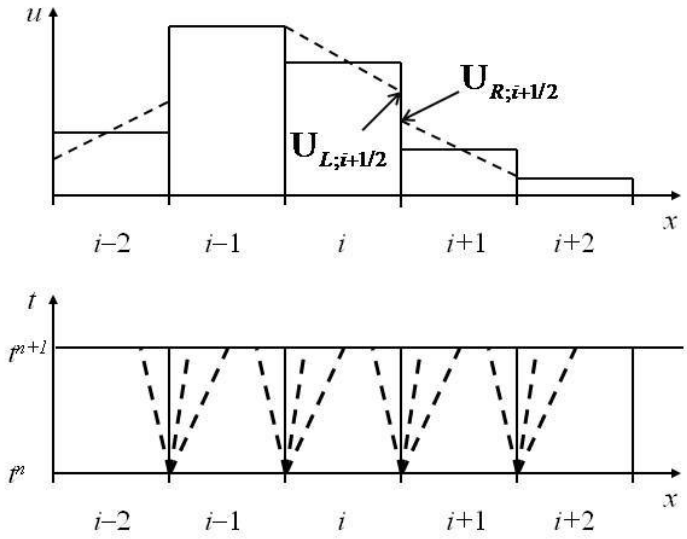


Fig. 3.13 schematically shows one stage of the second order Runge-Kutta method on a one-dimensional mesh. Here each zone is represented as a slab of fluid (solid lines) with a piecewise linear profile (dashed lines) in the top panel. The bottom panel shows the evolution in space and time of the Riemann problems at zone boundaries.

The observant reader may have noticed that the sequence of operations in eqns. (3.52), (3.53) and (3.54) represents a very complicated way of obtaining the reconstructed vector of slopes $\overline{\Delta \mathrm{U}}_i$ within each zone. A simpler alternative exists which is often used. If our goal is simply to achieve second order accuracy, this alternative is indeed computationally more efficient. Thus we write out the components of $\bar{\mathrm{U}}_i$ and $\overline{\Delta \mathrm{U}}_i$ explicitly so that we have $\bar{\mathrm{U}}_i \equiv \left(\bar{\mathrm{u}}_i^1, \bar{\mathrm{u}}_i^2, ..., \bar{\mathrm{u}}_i^M \right)^T$ and $\overline{\Delta \mathrm{U}}_i \equiv \left(\overline{\Delta \mathrm{u}}_i^1, \overline{\Delta \mathrm{u}}_i^2, ..., \overline{\Delta \mathrm{u}}_i^M \right)^T$. Recall that in each zone "*i*" $\bar{\mathrm{U}}_i$ is an *M*-component vector so that $\bar{\mathrm{u}}_i^k$ is the k^{th} component of that vector. We can then write

$$\overline{\Delta \mathrm{u}}_i^m = \mathit{Limiter}\left(\bar{\mathrm{u}}_{i+1}^m - \bar{\mathrm{u}}_i^m, \bar{\mathrm{u}}_i^m - \bar{\mathrm{u}}_{i-1}^m \right) \qquad \text{for } m = 1, ..., M \tag{3.57}$$

In other words, we reconstruct $\overline{\Delta \mathrm{U}}_i$ by applying the limiter componentwise to the vector of conserved variables. We call this *limiting on the conserved variables*. Notice that Sub-section 3.4.1 indeed informs us that the correct variables that one should reconstruct are in fact the characteristic variables. Even so, the componentwise reconstruction always

seems to have a way of working out for all of the hyperbolic systems that are likely to interest us in this book. Observe too that the scheme in Section 3.6.1 does not have this option of circumventing a characteristic projection because it requires the difference in characteristic variables, i.e. $\Delta w^m_{i+1/2}$ in eqn. (3.48), to construct the flux. Using this trick of componentwise limiting enables the scheme described in this Sub-section to match the speed of the scheme described in Sub-section 3.6.1.

TVD Property for the Improved Euler Runge-Kutta Scheme

To demonstrate the TVD property for the second order accurate scheme described in eqn. (3.51) we rewrite it as:

$$\bar{\mathrm{U}}_i^{(1)} = \bar{\mathrm{U}}_i^n - \frac{\Delta t}{\Delta x}\left(\bar{\mathrm{F}}^n_{i+1/2}\left(\bar{\mathrm{U}}^n\right) - \bar{\mathrm{F}}^n_{i-1/2}\left(\bar{\mathrm{U}}^n\right)\right)$$

$$\bar{\mathrm{U}}_i^{(2)} = \bar{\mathrm{U}}_i^{(1)} - \frac{\Delta t}{\Delta x}\left(\bar{\mathrm{F}}^{n+1/2}_{i+1/2}\left(\bar{\mathrm{U}}^{(1)}\right) - \bar{\mathrm{F}}^n_{i-1/2}\left(\bar{\mathrm{U}}^{(1)}\right)\right)$$

$$\bar{\mathrm{U}}_i^{n+1} = \frac{1}{2}\left(\bar{\mathrm{U}}_i^n + \bar{\mathrm{U}}_i^{(2)}\right)$$

Let "U" be an "*M*" component vector. By left-multiplying each of the above three equations by the matrix of left eigenvectors, "*L*", we can write the above three equations in terms of "*M*" scalar equations for the eigenweights as:

$$w_i^{m,n} \equiv l^m \bar{\mathrm{U}}_i^n$$

$$w_i^{m,(1)} = w_i^{m,n} - \frac{\Delta t}{\Delta x}\left(\mathrm{f}^m_{i+1/2}\left(w^{m,n}\right) - \mathrm{f}^m_{i-1/2}\left(w^{m,n}\right)\right)$$

$$w_i^{m,(2)} = w_i^{m,(1)} - \frac{\Delta t}{\Delta x}\left(\mathrm{f}^m_{i+1/2}\left(w^{m,(1)}\right) - \mathrm{f}^m_{i-1/2}\left(w^{m,(1)}\right)\right) \qquad \text{for } m = 1,\ldots,M$$

$$w_i^{m,n+1} = \frac{1}{2}\left(w_i^{m,n} + w_i^{m,(2)}\right)$$

$$\bar{\mathrm{U}}_i^{n+1} \equiv \sum_{m=1}^{M} w_i^{m,n+1} r^m$$

In other words, we start our time step at time t^n by obtaining "*M*" characteristic variables $w_i^{m,n}$ in each zone "*i*". We then evolve these characteristic variables as scalar advection, not as a system, using TVD-preserving fluxes. Observe that each of the two stages in the time-update above is TVD. (In practice, it is uneconomical to solve a hyperbolic problem

as described in this box; however, it yields useful insights.) For example, at the zone boundary "i+1/2" we evaluate $\mathrm{f}^{m}_{i+1/2}\left(w^{m,n}\right)$ for all the characteristic variables and use all such fluxes to obtain $w_i^{m,(1)}$ for each characteristic variable in each zone using the first stage. The TVD property of the first stage of our Runge-Kutta scheme then ensures that $\mathrm{TV}\left(w^{m,(1)}\right) \le \mathrm{TV}\left(w^{m,n}\right)$. Continuing the example, we then evaluate $\mathrm{f}^{m}_{i+1/2}\left(w^{m,(1)}\right)$ for each zone boundary "i+1/2" and each characteristic field. Using the second stage, we obtain $w_i^{m,(2)}$ for all the characteristic variables in all the zones "i". Since we used a TVD-preserving flux, we again have $\mathrm{TV}\left(w^{m,(2)}\right) \le \mathrm{TV}\left(w^{m,(1)}\right)$. For each zone "$i$" we can now obtain $w_i^{m,n+1}$ for all the characteristic variables as a convex combination of $w_i^{m,n}$ and $w_i^{m,(2)}$. The fact that it is a convex combination plays an important role in our proof of the TVD property. We use $w_i^{m,n+1}$ to obtain the final updated state $\bar{\mathrm{U}}_i^{n+1}$, thus completing the entire time-update. Fascinatingly, the TVD property is not obtained in the conserved state, but rather in the characteristic variables because we can write:

$$\mathrm{TV}\left(w^{m,n+1}\right) \le \frac{1}{2}\left[\mathrm{TV}\left(w^{m,n}\right) + \mathrm{TV}\left(w^{m,(2)}\right)\right] \le \frac{1}{2}\left[\mathrm{TV}\left(w^{m,n}\right) + \mathrm{TV}\left(w^{m,(1)}\right)\right] \le \mathrm{TV}\left(w^{m,n}\right) \quad \forall\ m = 1,..,M$$

I.e., we see that the entire time-evolution of the semi-discrete form can be written entirely in characteristic variables, and it is the characteristic variables that do indeed satisfy a TVD property for the scheme in eqn. (3.51). Consequently, we realize why the characteristic variables are more fundamental to the solution of a linear hyperbolic system, and this insight goes over to non-linear hyperbolic systems. We now come to a deeper understanding of why we carried out the reconstruction on the characteristic variables in eqns. (3.52) to (3.54).

For scalar advection we see that if each of the stages in eqn. (3.51) is TVD then the whole scheme is TVD. For linear hyperbolic systems, we could define the total variation of the hyperbolic system as the sum of the total variation in all its characteristic variables, i.e.

$$\mathrm{TV}\left(\bar{\mathrm{U}}^n\right) \equiv \sum_{m=1}^{M} \mathrm{TV}\left(w^{m,n}\right)$$

Such a definition would allow us to assert $\mathrm{TV}\left(\bar{\mathrm{U}}^{n+1}\right) \le \mathrm{TV}\left(\bar{\mathrm{U}}^n\right)$. However, this is dissatisfying because the above definition of total variation would not be the same as taking the component-wise L_1 norm of $\bar{\mathrm{U}}_{i+1} - \bar{\mathrm{U}}_i$ and then summing over all zones "*i*". In fact, it can be shown that a total variation defined in the component-wise norm will definitely not have a TVD property of the form that we would like to have. It is also difficult to gainfully extend the TVD concept to non-linear hyperbolic systems. Likewise, as shown by Goodman & LeVeque (1985), the TVD property cannot be profitably extended to include second order accurate techniques for scalar conservation laws in multiple dimensions. Despite its many limitations, the TVD property continues to give us the fundamental insight that in order to evolve the solution of a hyperbolic system in a non-oscillatory fashion, the variation of the solution within a zone has to be restricted via some sort of non-linear hybridization.

Stepwise Description of the Runge-Kutta Scheme

Since the scheme described in this sub-section is very useful, we catalogue it in terms of steps that are suitable for numerical implementation. We only describe one stage of the Runge-Kutta scheme:

Step 1: Impose boundary conditions. Then carry out reconstruction. Either use eqns. (3.52) to (3.54) or use eqn. (3.57).

Step 2: Obtain left and right states from eqn. (3.55). Use the Riemann solver as a machine to obtain the flux from eqn. (3.56).

Step 3: Use the appropriate update stage in either one of eqn. (3.50) or eqn. (3.51).

3.6.3) Predictor-Corrector Formulation

Observe that eqn. (3.50), the first stage in the two-stage Runge-Kutta scheme, only serves the role of advancing the solution to a time $t^n + \Delta t/2$ at which time we evaluate the fluxes that are actually used in the update eqn. (3.51). The final update

equation, i.e. eqn. (3.51), then carries the solution through the full timestep. As a result, we realize that there might be an alternative way of obtaining the left and right states for the Riemann solver in a way that already corresponds to the time $t^n + \Delta t/2$. By doing that we would be able to avoid the need to have separate predictor and corrector stages as in the Runge-Kutta scheme. Indeed we have already calculated the slope information at time t^n within each zone, which allows us to evolve the solution within the zone for a short amount of time. Thus within a zone "*i*" we can write

$$\frac{\Delta \mathrm{U}_i}{\Delta t} + \mathrm{A}\frac{\overline{\Delta \mathrm{U}}_i^n}{\Delta x} = 0 \tag{3.58}$$

so that we can evolve the solution for a time $\Delta t/2$ within zone "*i*" using the slope $\overline{\Delta \mathrm{U}}_i^n$ that can be obtained by applying the characteristic limiter from eqns. (3.52) to (3.54) to the mesh function at time t^n. A similar equation can be written for the time-evolution in zone "*i*+1". Eqn. (3.55) gives us the left and right states at the zone boundary "*i*+1/2" at time t^n. We would actually like to have those states at that zone boundary at time $t^n + \Delta t/2$. This is easily done by drawing on eqn. (3.58) and its analogue in zone "*i*+1". We first consider the left state and work through it in detail. Thus we can write

$$\begin{aligned} \mathrm{U}_{L;i+1/2}^{n+1/2} &\equiv \bar{\mathrm{U}}_i^n + \frac{1}{2}\,\overline{\Delta \mathrm{U}}_i^n + \frac{1}{2}\Delta t\left(\frac{\Delta \mathrm{U}_i}{\Delta t}\right) \\ &= \bar{\mathrm{U}}_i^n + \frac{1}{2}\,\overline{\Delta \mathrm{U}}_i^n - \frac{1}{2}\frac{\Delta t}{\Delta x}\,\mathrm{A}\,\overline{\Delta \mathrm{U}}_i^n \end{aligned} \tag{3.59}$$

Using eqn. (3.54) in (3.59) we get

$$\mathrm{U}_{L;i+1/2}^{n+1/2} = \bar{\mathrm{U}}_i^n + \frac{1}{2}\,\overline{\Delta \mathrm{U}}_i^n - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_{m=1}^{M}\lambda^m\,\Delta w_i^m\,r^m \tag{3.60}$$

An analogous exercise for the right state is given by

$$\begin{aligned} \mathrm{U}^{n+1/2}_{R;i+1/2} &\equiv \bar{\mathrm{U}}^{n}_{i+1} - \frac{1}{2}\,\overline{\Delta \mathrm{U}}^{n}_{i+1} + \frac{1}{2}\Delta t\left(\frac{\Delta \mathrm{U}_{i+1}}{\Delta t}\right) \\ &= \bar{\mathrm{U}}^{n}_{i+1} - \frac{1}{2}\,\overline{\Delta \mathrm{U}}^{n}_{i+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\,\mathrm{A}\,\overline{\Delta \mathrm{U}}^{n}_{i+1} \end{aligned} \tag{3.61}$$

and it yields

$$\mathrm{U}^{n+1/2}_{R;i+1/2} = \bar{\mathrm{U}}^{n}_{i+1} - \frac{1}{2}\,\overline{\Delta \mathrm{U}}^{n}_{i+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_{m=1}^{M}\lambda^{m}\,\Delta w^{m}_{i+1}\,r^{m} \tag{3.62}$$

Eqns. (3.60) and (3.62) can be thought of as the *predictor step* for this scheme. The figure in the box at the end of Section 3.3 illustrates how the predicted values at the zone boundary "i+1/2", i.e. $\mathrm{U}^{n+1/2}_{R;i-1/2}$ and $\mathrm{U}^{n+1/2}_{L;i+1/2}$, are obtained within the simpler context of scalar advection. If we think of the Riemann solver as a machine that takes two states, one from the left and another from the right, and produces a resolved flux then our single stage time-update is given by

$$\bar{\mathrm{U}}^{n+1}_{i} = \bar{\mathrm{U}}^{n}_{i} - \frac{\Delta t}{\Delta x}\left(\mathrm{F}_{RS}\left(\,\mathrm{U}^{n+1/2}_{L;i+1/2}\,,\mathrm{U}^{n+1/2}_{R;i+1/2}\,\right) - \mathrm{F}_{RS}\left(\,\mathrm{U}^{n+1/2}_{L;i-1/2}\,,\mathrm{U}^{n+1/2}_{R;i-1/2}\,\right)\right) \tag{3.63}$$

Eqn. (3.63) can be thought of as the *corrector step* that gives us the full time update. Such a scheme was first proposed by Colella (1985) and was also used in Colella & Woodward (1984).

The present method offers the same stability properties as the previous Runge-Kutta method. Both methods work with the same CFL restriction. However, the reconstructed profiles only have to be evaluated once for the present method. Likewise, we only need to make one evaluation of the Riemann problem for the method described in this sub-section. For non-linear hyperbolic systems, the Riemann solver can dominate the cost of the scheme. As a result, the present scheme has a favorable computational efficiency relative to the Runge-Kutta method. We see, however, that the left and right

states, i.e. eqns. (3.60) and (3.62), require a more intricate construction than the left and right states in eqn. (3.55) for the Runge-Kutta method. Thus, in a way of speaking, there is a trade-off between simplicity of programming, which favors the Runge-Kutta method, and computational efficiency, which favors the present method. In Chapter 7 we will present variants of these two methods (at second and higher orders of accuracy) for non-linear hyperbolic systems. We will see that the same trade-offs prevail even in that situation.

There is a small latitude that one can draw on in the definition of the left and right states that go into the Riemann solver. Colella (1985) utilized that latitude to obtain expressions that are slightly different from our expressions in eqns. (3.60) and (3.62). Thus let λ^1 and λ^M be the smallest and largest eigenvalues respectively of the characteristic matrix "A" (i.e. we assume that the eigenvalues are ordered from smallest to largest). Then the left state in eqn. (3.55) was taken to be a spatial average over all the waves that reach the zone boundary from the left zone in a time Δt. Similarly, the right state in eqn. (3.55) was taken to be a spatial average over all the waves that reach the zone boundary from the right zone in a time Δt. As a result, the expressions preferred by Colella (1985) are

$$\mathrm{U}_{L;i+1/2}^{n+1/2} = \bar{\mathrm{U}}_i^n + \frac{1}{2}\left(1 - \max\left(\lambda^M, 0\right)\frac{\Delta t}{\Delta x}\right)\overline{\Delta \mathrm{U}}_i^n - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_{m=1}^{M}\lambda^m\,\Delta w_i^m\,r^m \tag{3.64}$$

$$\mathrm{U}_{R;i+1/2}^{n+1/2} = \bar{\mathrm{U}}_{i+1}^n - \frac{1}{2}\left(1 + \min\left(\lambda^1, 0\right)\frac{\Delta t}{\Delta x}\right)\overline{\Delta \mathrm{U}}_{i+1}^n - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_{m=1}^{M}\lambda^m\,\Delta w_{i+1}^m\,r^m \tag{3.65}$$

In practice, it has been found that eqns. (3.64) and (3.65) are a little more stabilizing than eqns. (3.60) and (3.62).

Stepwise Description of the Predictor-Corrector Scheme

Since the Predictor-Corrector scheme described in this sub-section is very useful, we catalogue it in terms of steps that are suitable for numerical implementation:

Step 1: Impose boundary conditions. Then carry out reconstruction. Either use eqns. (3.52) to (3.54) or use eqn. (3.57).

Step 2: Obtain left and right states from eqns. (3.60) and (3.62). This is the predictor step.

Step 3: Use the Riemann solver as a machine to obtain the numerical flux from eqn. (3.56).

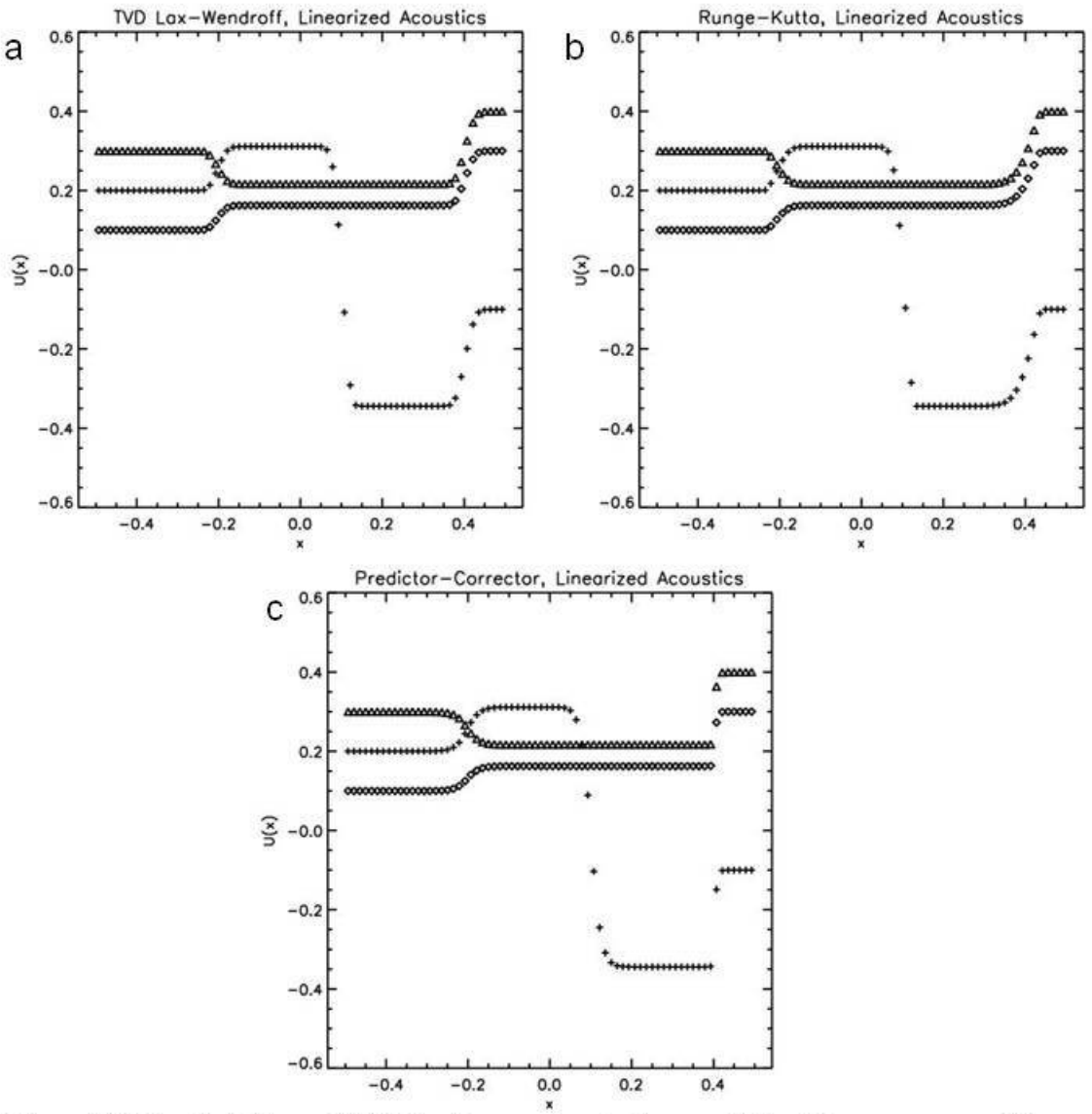
Step 4: Update using eqn. (3.63). This is the corrector step.

3.6.4) Numerical Results from the Previous Three Schemes

As our numerical example we take the linearized, one-dimensional Euler system that was catalogued in the box at the end of Section 3.4. The characteristic matrix in our numerical example is defined by setting $\rho_0 = 1$, $\mathrm{v}_{x0} = 0.25$ and $\mathrm{c}_0 = 0.75$. We solve a Riemann problem numerically using the three schemes that we designed in the previous three sub-sections. We take the left and right states to be $\mathrm{U}_L = \left(0.2, 0.3, 0.1\right)^T$ and $\mathrm{U}_R = \left(-0.1, 0.4, 0.3\right)^T$ respectively. Note that these left and right states can be interpreted as fluctuations that are applied to the mean values of the density, x-velocity and pressure in the Euler equations. As fluctuations go, the present fluctuations are quite large and would provoke non-linear effects in the full Euler equation. However, we are only solving a linear problem that is derived from the Euler equations and so it is acceptable to have these left and right states for our Riemann problem. The Riemann problem was initialized in the middle of a 70 zone mesh which spans $\left[-0.5, 0.5\right]$ along the *x*-axis. The MC limiter was applied to the characteristic variables for the schemes in Sub-sections 3.6.1 and 3.6.2. The MC limiter was found to be too compressive for the scheme described in Sub-section 3.6.3 and so we switched to a minmod limiter. A CFL number of 0.6 was used for all three schemes and they were evolved to a final time of 0.4.

Fig. 3.14a, 3.14b and 3.14c show the results from the schemes in Sub-sections 3.6.1, 3.6.2 and 3.6.3 respectively. The crosses show the fluctuations in the density, the triangles show the fluctuations in the velocity and the diamonds show the fluctuations in the pressure. We see that the Riemann problem has been resolved by all three schemes

into a sequence of three simple waves. The leftmost discontinuity is a left-going sound wave and generates changes in the density, x-velocity and pressure. The relative sizes of the jumps produced in the variables across the left-going sound wave are proportional to the components of the first right eigenvector r^1 . The middle wave is a contact discontinuity. It does not produce changes in the x-velocity or the pressure; its changes are restricted to producing a jump in the density variable. This can also be seen by an examination of the second right eigenvector r^2 . The rightmost discontinuity is a right-going sound wave and again generates jumps in all three variables that are proportional to the third right eigenvector r^3. (The eigenvectors are catalogued in Sub-section 1.5.2.) The only noticeable difference between Figs. 3.14a and 3.14b arises in the treatment of the right-going sound wave where we see that the profile is smeared a little more in Fig. 3.14b than in Fig. 3.14a. This can be understood because the scheme with Runge-Kutta timestepping applies the Riemann problem twice in the course of a time step, thus producing a slightly greater dissipation in the fastest propagating wave. In practice, the full, non-linear Euler system displays a self-steepening of sound waves. Consequently, this extra dissipation in the rightward propagating sound wave would be counteracted by the tendency of the hyperbolic system to cause its sound waves to self-steepen. Fig. 3.14c shows that the waves are also crisply preserved in the scheme from Sub-section 3.6.3. This time, the central wave is slightly smeared because of the use of the minmod limiter.

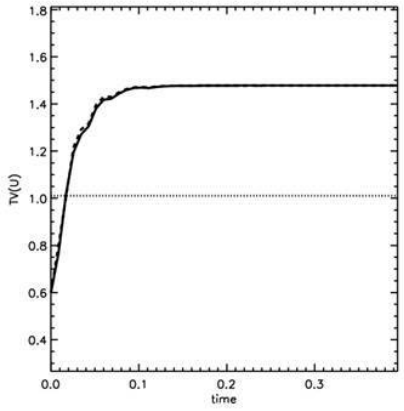


Figs. 3.14a, 3.14b and 3.14c show simulations of the Riemann problem for the linearized acoustics equations using the TVD Lax-Wendroff scheme, the two-stage Runge-Kutta scheme and the predictor corrector scheme respectively. The crosses show the fluctuations in the density; the triangles show the fluctuations in the x-velocity and the diamonds show the fluctuations in the pressure. A CFL number of 0.6 was used for all schemes.

TVD Property for Hyperbolic Systems?

In the box at the end of Sub-section 3.6.2 we documented that the improved Euler approximation has a TVD property. We observed that the Runge-Kutta scheme described

there only retains the TVD property for linear hyperbolic systems when it is formulated in characteristic variables. When formulated in conserved variables, the property does not hold. We now catalogue the time evolution of the total variation for the problem described in this sub-section. The dotted curve shows the total variation as measured in the characteristic variables. We see that the improved Euler approximation does an excellent job of preserving the total variation in the characteristic variables. We had also mentioned that the same property does not hold in conserved variables. The solid and dashed curves show the total variation measured for the conserved variables when the improved Euler and modified Euler approximations are used. Consistent with our expectation, we see that the total variation does increase for either of those two Runge-Kutta schemes. In fact, there is practically no difference between them.



The figure to the left shows the evolution of the total variation as a function of time for the simulation that was described in this sub-section. The dotted curve shows the evolution of the total variation as measured in characteristic variables. The solid and dashed curves practically overlie each other and pertain to the total variation measured in the conserved variables when the improved Euler and modified Euler Runge-Kutta schemes are used.

References

Arora, M. and Roe, P.L., *On post-shock oscillations due to shock-capturing schemes in unsteady flow*, Journal of Computational Physics, 130 (1997) 1

Balsara,D.S., Altmann, C., Munz, C.-D., Dumbser, M., *A Sub-cell Based Indicator for Troubled Zones in RKDG Schemes and a Novel Class of Hybrid RKDG+HWENO Schemes*, Journal of Computational Physics, 226 (2007) 586-620

Colella, P. and Woodward, P.R., *The piecewise parabolic method (PPM) for gas-dynamical simulations*, Journal of Computational Physics, 54 (1984) 174-201

Colella, P., *A direct Eulerian MUSCL scheme for gas dynamics*, SIAM Journal of Scientific and Statistical Computing, 6 (1985) 104-117

Dumbser, M., Balsara, D.S., Toro, E.F. and Munz, C.D., *A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes*, Journal of Computational Physics, 227 (2008) 8209-8253

Fromm, J.E., A *Method for Reducing Dispersion in Convective Difference Schemes*, Journal of Computational Physics, 3 (1968) 176

Godunov, S.K., *A difference method for the numerical calculation of discontinuous solutions of the equations of hydrodynamics*, Mat. Sb., 47 (1959) 271-306

Goodman, J.B. and LeVeque, R.J., *On the accuracy of stable schemes for 2D scalar conservation laws*, Math. Comp., 45 (1985) 15

Harten, A., *High resolution schemes for conservation laws*, Journal of Computational Physics, 49 (1983) 357-393

Hedstrom, G.W., *Non-reflecting boundary conditions for nonlinear hyperbolic systems*, Journal of Computational Physics, 30 (1979) 222

Jameson, A., in *Transonic, shock and multidimensional flows: Advances in scientific computing*, edited by R. Meyer (Academic press, New York) (1982) 37

Lax, P.D., *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM Regional Conference in Applied Mathematics # 11 (1972)

Liu, Q. & Vasilyev, O.V., *Nonreflecting boundary conditions based on nonlinear multidimensional characteristics*, Int. J. Numer. Meth. Fl., 62 (2010) 24-55

Oleinik, O., *Discontinuous solutions of non-linear differential equations,* Amer. Math. Soc. Transl. Ser. 2(26) (1957) 95-172

Osher, S. and Chakravarthy, S., *High resolution schemes and the entropy condition*, SIAM J. Num. Anal., 21 (1984) 995-984

Riemann, B., *Über die fortpflanzung ebener Luftwellen von endlicher Schwingungsweite*, Abhandlungen der Gesselschaft der Wissenschaften zun Göttingen, Mathematisch-physikalische Klasse, 8 (1860) 43

Roe, P.L., *Approximate Riemann solver, parameter vectors and difference schemes*, Journal of Computational Physics, 43 (1981) 357-372

Roe, P.L., *Characteristic-based schemes for the Euler equations*, Ann. Rev. Fluid Mech., 18 (1986) 337

Spiteri, R.J. & Ruuth**,** S.J., *Non-linear evolution using optimal fourth-order strong-stability-preserving Runge-Kutta methods*, Mathematics and Computers in Simulation 62 (2003) 125-135

Shu, C.-W. and Osher, S. J., *Efficient implementation of essentially non-oscillatory shock capturing schemes*, Journal of Computational Physics, 77 (1988) 439-471

Shu, C.-W., *Total variation-diminishing time discretizations*, SIAM Journal of Scientific and Statistical Computing, 9 (1988) 1073-1084

Sutherland, J.C. and Kennedy, C.A., *Improved boundary conditions for viscous, reacting, compressible flows*, Journal of Computational Physics, 191 (2003) 502-524

Sweby,P.K., *High resolution schemes using flux-limiters for hyperbolic conservation laws*, SIAM Journal of Numerical Analysis, 21 (1984) 995-1011

Tadmor, E, *Convenient total variation diminishing conditions for nonlinear difference schemes*, SIAM J. Num. Anal., 25 (1988), 1002-1014

Thompson, K.W., *Time-dependent boundary conditions for hyperbolic systems, II*, Journal of Computational Physics, 89 (1990) 439-461

Titarev, V.A., Toro, E.F., ADER: arbitrary high order Godunov approach, Journal of Scientific Computing 17 (1-4) (2002) 609-618

Toro, E.F., *Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction*, Springer (2009)

van Albada, G.D., van Leer, B. and Roberts, W.W., *A comparative study of computational methods in cosmic gas dynamics*, Astronomy and Astrophysics, 108 (1982) 76

Van Leer, B, (1974), *Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second order scheme*, Journal of Computational Physics, 14, p361-70

van Leer, B., *Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection*, Journal of Computational Physics, 23 (1977) 276-299

van Leer, B., *Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method*, Journal of Computational Physics, 32 (1979) 101

Yee, H.C., Beam, R.M. and Warming, R.F., AIAA Journal, 20 (1982) 1203

Problem Set

3.1) Use right differences, $\overline{\Delta u}_i^n = \bar{u}_{i+1}^n - \bar{u}_i^n$, in eqn. (3.7) to show that it reduces to the Lax-Wendroff scheme from Sub-section 2.7.3. Can you show that the scheme is not positivity preserving?

3.2) Use centered differences, $\overline{\Delta u}_i^n = \frac{1}{2}\left(\bar{u}_{i+1}^n - \bar{u}_{i-1}^n\right)$, in eqn. (3.7) to show that it reduces to the Fromm scheme which is given by

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\mu}{4}\left(\bar{u}_{i+1}^n + 3\,\bar{u}_i^n - 5\,\bar{u}_{i-1}^n + \bar{u}_{i-2}^n\right) + \frac{\mu^2}{4}\left(\bar{u}_{i+1}^n - \bar{u}_i^n - \bar{u}_{i-1}^n + \bar{u}_{i-2}^n\right)$$

Use eqn. (2.46) to show that this scheme is second order accurate. Can you show that the scheme is not positivity preserving? Notice that while the centered difference might have seemed like the most symmetrical choice, it does not yield a symmetrical scheme, nor is its stencil symmetrical.

3.3) Use left differences, $\overline{\Delta u}_i^n = \bar{u}_i^n - \bar{u}_{i-1}^n$, in eqn. (3.7) to show that it reduces to the Beam-Warming scheme given by

$$\bar{u}_i^{n+1} = \bar{u}_i^n - \frac{\mu}{2}\left(3\,\bar{u}_i^n - 4\,\bar{u}_{i-1}^n + \bar{u}_{i-2}^n\right) + \frac{\mu^2}{4}\left(\bar{u}_i^n - 2\,\bar{u}_{i-1}^n + \bar{u}_{i-2}^n\right)$$

Using eqn. (2.46), show that this scheme is second order accurate. As in the previous problem, notice that the choice of a left difference does not yield a symmetrical scheme. Can you show that the scheme is not positivity preserving? Exercises 3.1, 3.2 and 3.3 taken together display the full set of one-stage, time-explicit, second order schemes that can be designed with a compact stencil. We see that the latter two schemes are not even symmetrical.

3.4) For a continuous function the total variation is defined by $\mathrm{TV}(f) = \int_{-\infty}^{\infty} \left| f'(x) \right| dx$. Extend the concept for discontinuous functions. Find the total variation for the following functions:

a) $f(x) = e^{-(x/0.1)^2}$

b) $f(x) = \sin(x)\ \forall\ x \in [-\pi, \pi]$ and $f(x) = 0$ for all other values of *x*.

c) $f(x) = 1\ \forall\ x \in [-0.05, 0.05]$ and $f(x) = 0$ for all other values of *x*.

3.5) In the text, we asserted that eqn. (3.21) was sufficient to ensure the TVD property for the scheme in eqn. (3.19). We now prove the claim. Consider the function $f(\mu) = \mu + \frac{\mu}{2}(1-\mu)D$ for $0 \le \mu \le 1$ and some value $D = \frac{\phi(\theta_i)}{\theta_i} - \phi(\theta_{i-1})$. Show that requiring $0 \le f(\mu) \le 1$ is equivalent to requiring that $-\frac{2}{1-\mu} \le D \le \frac{2}{\mu}$. (Hint: Do this by first subtracting μ and then dividing by $\frac{\mu}{2}(1-\mu)$.) Consequently, for $0 \le \mu \le 1$, we realize that $-2 \le D \le 2$ is sufficient for guaranteeing the TVD property. This problem also highlights the fact that $\frac{\phi(\theta_i)}{\theta_i} - \phi(\theta_{i-1})$ can lie within a larger range than the one asserted in eqn. (3.21). The larger range can be exploited to obtain a sharper limiter, as was done in the box at the end of Section 3.3.

3.6) Assume a symmetrical mesh function around the zone boundary "*i*–1/2" so that $\bar{u}_{i-2} = -b$, $\bar{u}_{i-1} = -a$, $\bar{u}_i = a$ and $\bar{u}_{i+1} = b$. To make the example concrete assume $0 \le a < b$. Obtain the limited, undivided difference $\overline{\Delta u_i}$ in zone "*i*" by using the fact that

$$\overline{\Delta u_i} = \phi\left(\frac{\bar{u}_i - \bar{u}_{i-1}}{\bar{u}_{i+1} - \bar{u}_i}\right)(\bar{u}_{i+1} - \bar{u}_i)$$

Similarly obtain $\overline{\Delta \mathrm{u}}_{i-1}$. By specializing the values in the formulae using *a* and *b* , show that our requirement that the slopes be symmetrical about the zone boundary "*i*–1/2" gives us $\overline{\Delta \mathrm{u}}_{i} = \overline{\Delta \mathrm{u}}_{i-1}$. By asserting this equality, show that

$$\frac{\phi\left(\frac{2a}{b-a}\right)}{\left(\frac{2a}{b-a}\right)} = \phi\left(\frac{b-a}{2a}\right)$$

Since *a* and *b* are general values, realize that we have just proved the statement in eqn. (3.24).

3.7) Substitute eqn. (3.31) into eqn. (3.26) and verify that it is the correct solution of the linear PDE that satisfies the initial conditions given by the smooth and differentiable vector function $\mathrm{U}_0(x)$.

3.8) Use integration by parts and Fig. 3.9 to show that eqn. (3.32) is the appropriate discontinuous solution of eqn. (3.26).

3.9) Show that the expressions in eqn. (3.42) are equivalent to their counterparts in eqn. (3.41). This is most easily done by substituting the definitions from eqn. (3.43) in eqn. (3.42) and simplifying the resulting expressions. You will also have to use the definition $\alpha^m \equiv l^m\left(\mathrm{U}_R - \mathrm{U}_L\right)$.

3.10) Assume an x-directional, non-zero, initial magnetic field in the one-dimensional MHD system. Eliminate all fluctuations in the density, pressure, x-velocity and x-component of the magnetic field. Also eliminate the z-components of the velocity and magnetic field altogether. Show that the fluctuations in the y-components of the velocity and magnetic field satisfy the 2×2 system given by

$$\frac{\partial}{\partial t}\begin{pmatrix} \mathrm{v}_y \\ \mathrm{B}_y \end{pmatrix} + \begin{pmatrix} \mathrm{v}_{x0} & -\dfrac{\mathrm{B}_{x0}}{4\,\pi\,\rho_0} \\ -\mathrm{B}_{x0} & \mathrm{v}_{x0} \end{pmatrix} \frac{\partial}{\partial x}\begin{pmatrix} \mathrm{v}_y \\ \mathrm{B}_y \end{pmatrix} = 0$$

Here ρ_0 , v_{x0} and B_{x0} are the unperturbed density, x-velocity and x-component of the magnetic field. Show that the eigenvalues are given by $\lambda^1 = \mathrm{v}_{x0} - \mathrm{v}_\mathrm{A}$ and $\lambda^2 = \mathrm{v}_{x0} + \mathrm{v}_\mathrm{A}$ where $\mathrm{v}_\mathrm{A} = \left|\mathrm{B}_{x0}\right| \Big/ \sqrt{4\,\pi\,\rho_0}$ is called the Alfven speed. Find the left and right eigenvectors and obtain the resolved flux for this system using steps that parallel the steps in the box at the end of Section 3.4.

Computer Exercises

3.1) Using eqns. (3.7) and (3.8) reproduce the results given in Fig. 3.5

3.2) Using eqns. (3.7) and the MC limiter that was given at the end of Section 3.2, reproduce the results given in Fig. 3.6

3.3) Apply the Superbee limiter to produce results that are analogous to Figs. 3.5 and 3.6. What can you say about the quality of the solution produced by the superbee limiter?

3.4) Apply the Superbee limiter to the numerical advection of the function sin (2 π x) on the unit interval. Use periodic geometry and advect the sine wave for several cycles. On a small enough mesh you should see that the sine wave starts turning into a square wave, thus illustrating the overcompressive nature of the limiter.

3.5) Reproduce the results from Figs. 3.14a and 3.14b by coding up both the schemes from Section 3.6.

3.6) For the 2×2 hyperbolic system defined in problem 3.10 above, set $\rho_0 = 1$, $\mathrm{v}_{x0} = 0.5$ and $\mathrm{B}_{x0} = \sqrt{4\pi}$. Set up a 100 zone mesh spanning $\left[-0.5, 0.5\right]$ along the *x*-axis. At x=0

initialize a Riemann problem with $U_L = (0.1, 0.3)$ and $U_R = (-0.2, 0.2)$. Use a CFL number of 0.6 and evolve the Riemann problem to a final time of 0.3. Try both the schemes from Section 3.5.