Unit 3. Introduction to Computational Fluid Dynamics (CFD)

Lecture 313: Finite-volume methods: Riemann problem

Reference book:

“Introduction to Computational Astrophysical Hydrodynamics” by Zingale.
http://bender.astro.sunysb.edu/hydro_by_example/CompHydroTutorial.pdf

W. Banda-Barragán, 2023

Introduction to CFD: Advection and the finite-volume method

In these notes, we will typically use a *finite-volume* discretization. Here we explore this method for the advection equation. First we rewrite the advection equation in *conservation form*:

$$a_t + [f(a)]_x = 0 \tag{5.1}$$

where $f(a) = ua$ is the flux of the quantity a. In conservation form, the time derivative of a quantity is related to the divergence of its flux.

Recall that in the finite-volume discretization, $\langle a \rangle_i$ represents the average of $a(x,t)$ over the interval $x_{i-1/2}$ to $x_{i+1/2}$, where the half-integer indexes denote the zone edges (i.e. $x_{i-1/2} = x_i - \Delta x/2$).

Remember: $$\langle f \rangle_i = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)dx \sim f(x_i)$$

Introduction to CFD: Advection and the finite-volume method

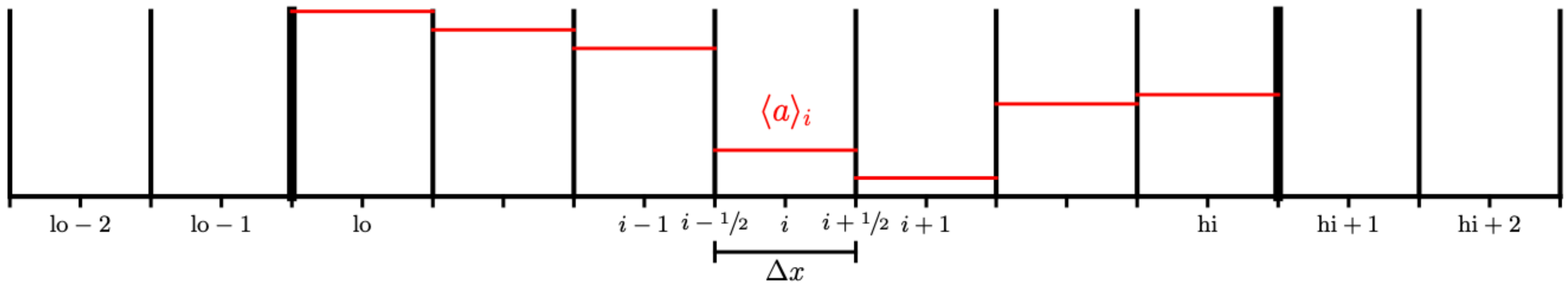


Figure 5.1: A finite-volume grid running from lo, . . . , hi, with two ghost cells at each end.

Figure 5.1 shows an example of such a grid with 2 ghost cells at each end. (For simplicity of notation, we drop the $\langle\rangle$ going forward). To discretize Eq. 5.1, we integrate it over a zone, from $x_{i-1/2}$ to $x_{i+1/2}$, normalizing by the zone width, Δx:

$$\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} a_t \, dx = -\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial}{\partial x} f(a) \, dx \tag{5.2}$$

Introduction to CFD: Advection and the finite-volume method

$$\frac{\partial}{\partial t} a_i = -\frac{1}{\Delta x} \left\{ [f(a)]_{i+1/2} - [f(a)]_{i-1/2} \right\} \tag{5.3}$$

This is an evolution equation for the zone-average of a, and shows that it updates in time based on the fluxes through the boundary of the zone.

We now have a choice on how to proceed with the time-discretization:

1. We can discretize $\partial a_i / \partial t$ directly as $(a_i^{n+1} - a_i^n)/\Delta t$. Then to achieve second-order accuracy, we need to evaluate the righthand side of Eq. 5.3 at the midpoint in time $(n + 1/2)$. This gives rise to a predictor-corrector method (as described in [24]. Sometimes this is called a characteristic tracing method (a name which will be more clear when we discuss the Euler equations).

2. We can recognize that with the spatial discretization done, our PDE has now become an ODE, and we can use standard ODE methods (like Runge-Kutta) to integrate the system in time. This is a method-of-lines integration.

Introduction to CFD: Second-order predictor-corrector scheme

$$\frac{\partial}{\partial t} a_i = -\frac{1}{\Delta x}\left\{[f(a)]_{i+1/2} - [f(a)]_{i-1/2}\right\}$$

We discretize Eq. 5.3 in time by evaluating the righthand side at the midpoint in time—this gives a centered-difference in time, which is second-order accurate:

$$\frac{a_i^{n+1} - a_i^n}{\Delta t} = -\frac{[f(a)]_{i+1/2}^{n+1/2} - [f(a)]_{i-1/2}^{n+1/2}}{\Delta x} \tag{5.4}$$

To evaluate the fluxes at the half-time, we need the state at the half-time, that is, we do :

$$[f(a)]_{i+1/2}^{n+1/2} = f(a_{i+1/2}^{n+1/2}) \quad . \tag{5.5}$$

Introduction to CFD: Second-order predictor-corrector scheme

We construct a second-order accurate approximation to $a_{i+1/2}^{n+1/2}$ by Taylor expanding the data in the cell to the interface. The construction of the interface state at the midpoint in time is the prediction and the conservative update in the correction here.

Notice that for each interface, there are two possible interface states we can construct—one using the data to the left of the interface (which we will denote with a "L" subscript) and the other using the data to the right of the interface (denoted with an "R" subscript)—see Figure 5.2.

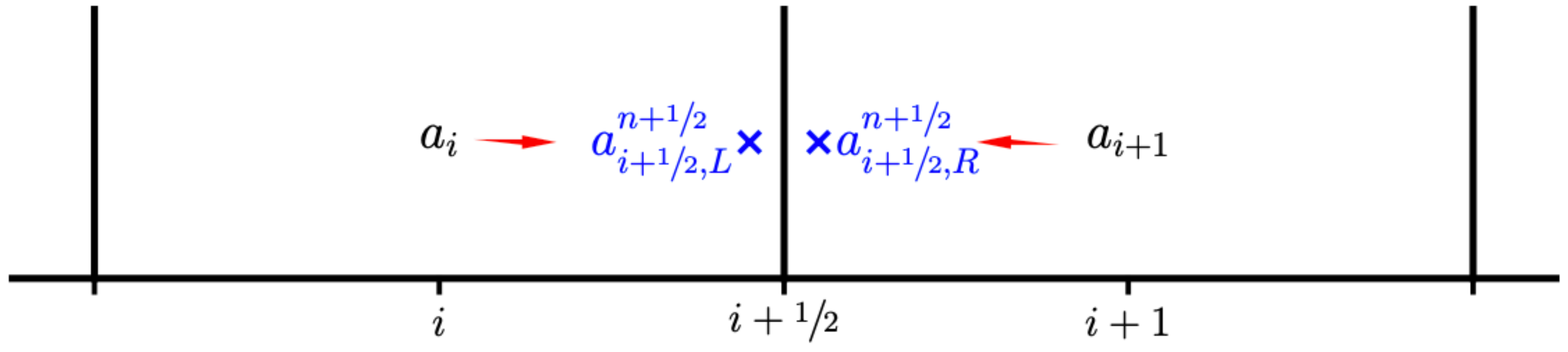


Figure 5.2: The left and right interface state at the $i + 1/2$ interface. Here, the left state, $a_{i+1/2,L}^{n+1/2}$, was predicted to the interface from the zone to the left of the interface, using a_i, and the right state, $a_{i+1/2,R}^{n+1/2}$, was predicted to the interface from the zone to the right, using a_{i+1}.

Introduction to CFD: Second-order predictor-corrector scheme

These states are:

$$\begin{aligned}
a_{i+1/2,L}^{n+1/2} &= a_i^n + \frac{\Delta x}{2}\left.\frac{\partial a}{\partial x}\right|_i + \frac{\Delta t}{2}\left.\frac{\partial a}{\partial t}\right|_i + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2) \\
&= a_i^n + \frac{\Delta x}{2}\left.\frac{\partial a}{\partial x}\right|_i + \frac{\Delta t}{2}\left(-u\left.\frac{\partial a}{\partial x}\right|_i\right) + \dots \\
&= a_i^n + \frac{\Delta x}{2}\left(1 - \frac{\Delta t}{\Delta x}u\right)\left.\frac{\partial a}{\partial x}\right|_i + \dots
\end{aligned} \tag{5.6}$$

$$\begin{aligned}
a_{i+1/2,R}^{n+1/2} &= a_{i+1}^n - \frac{\Delta x}{2}\left.\frac{\partial a}{\partial x}\right|_{i+1} + \frac{\Delta t}{2}\left.\frac{\partial a}{\partial t}\right|_{i+1} + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2) \\
&= a_{i+1}^n - \frac{\Delta x}{2}\left.\frac{\partial a}{\partial x}\right|_{i+1} + \frac{\Delta t}{2}\left(-u\left.\frac{\partial a}{\partial x}\right|_{i+1}\right) + \dots \\
&= a_{i+1}^n - \frac{\Delta x}{2}\left(1 + \frac{\Delta t}{\Delta x}u\right)\left.\frac{\partial a}{\partial x}\right|_{i+1} + \dots
\end{aligned} \tag{5.7}$$

Introduction to CFD: Riemann problem

A suitable estimate is needed for the slope of a that appears in these expressions (as $\partial a/\partial x$). We can approximate this simply as

$$\left.\frac{\partial a}{\partial x}\right|_i = \frac{a_{i+1} - a_{i-1}}{2\Delta x} \tag{5.8}$$

We can think of this method as reconstructing the function form of the data from the cell-average data in each cell using a piecewise linear polynomial. Don't be worried that this looks like FTCS—we'll do upwinding next.

We now have two states, $a^{n+1/2}_{i+1/2,L}$ and $a^{n+1/2}_{i+1/2,R}$ separated by an interface—this is called the *Riemann problem*. The solution to this will depend on the equation being solved, and results in a single state at the interface:

$$a^{n+1/2}_{i+1/2} = \mathcal{R}(a^{n+1/2}_{i+1/2,L}, a^{n+1/2}_{i+1/2,R}) \tag{5.9}$$

Introduction to CFD: Riemann problem

In our case, the advection equation simply propagates the state to the right (for $u > 0$), so the solution to the Riemann problem is to take the left state (this is another example of upwinding). That is we do:

$$\mathcal{R}(a_{i+1/2,L}^{n+1/2}, a_{i+1/2,R}^{n+1/2}) = \begin{cases} a_{i+1/2,L}^{n+1/2} & u > 0 \\ a_{i+1/2,R}^{n+1/2} & u < 0 \end{cases} \tag{5.10}$$

To complete the update, we use this interface state to evaluate the flux and update the advected quantity via Eq. 5.4 and 5.5.

$$\frac{a_i^{n+1} - a_i^n}{\Delta t} = -\frac{[f(a)]_{i+1/2}^{n+1/2} - [f(a)]_{i-1/2}^{n+1/2}}{\Delta x} \tag{5.4}$$

Remember:

$$[f(a)]_{i+1/2}^{n+1/2} = f(a_{i+1/2}^{n+1/2}) \ . \tag{5.5}$$

Introduction to CFD: BCs in the Riemann problem

Boundary conditions are implemented by filling the ghost cells outside each end of the domain based on data in the interior. Note that at the very left edge of the domain, the state $a^{n+1/2}_{\mathrm{lo}-1/2}$ requires the construction of states on the left and right. The left state at that interface, $a^{n+1/2}_{\mathrm{lo}-1/2,L}$ depends on the slope reconstructed in the $\mathrm{lo}-1$ ghost cell, $\partial a/\partial x|_{\mathrm{lo}-1}$. This in turn is constructed using a limited center-difference that will consider the value in the cell to the left, $\mathrm{lo}-2$. Therefore, we need two ghost cells at each end of the domain for this method—figure 5.3 illustrates this. Higher-order limiters may require even more ghost cells.

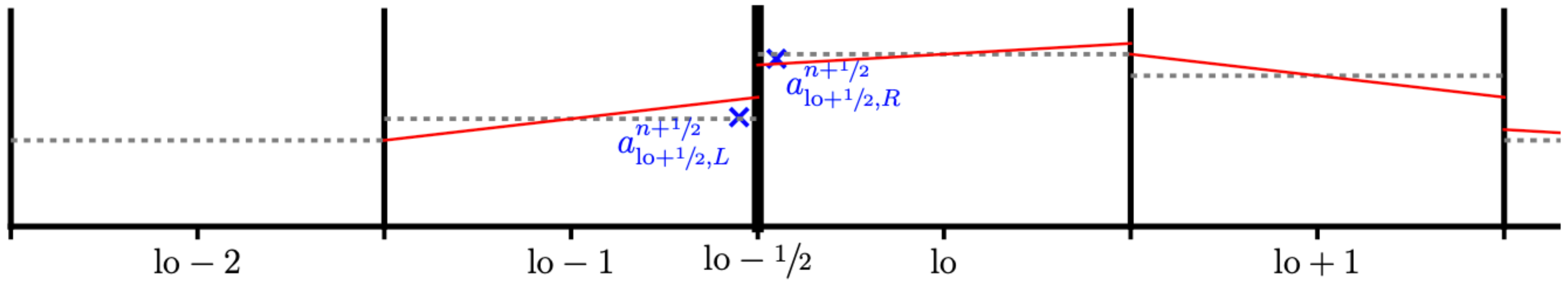
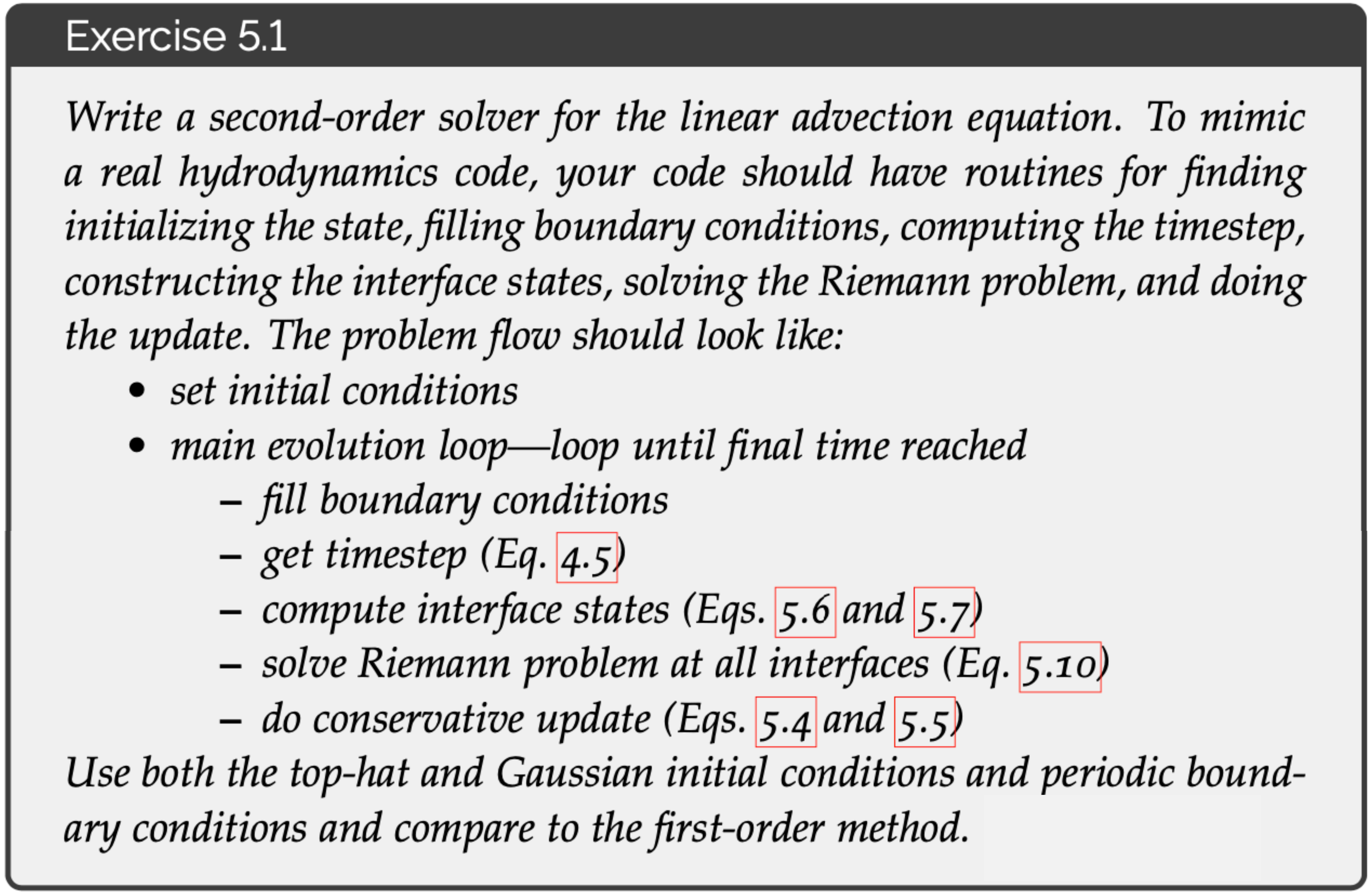


Figure 5.3: Reconstruction near the boundary, showing the need for two ghostcells. Here we see the left and right state at the left physical boundary of the domain (marked as $\mathrm{lo}-1/2$). The gray dotted lines are the piecewise constant cell averages and the red lines are the reconstructed slopes. Note that we need the slope in $\mathrm{lo}-1$ to get the left interface state at $\mathrm{lo}-1/2$, and that slope in turn needed the data in zone $\mathrm{lo}-2$ to construct a centered-difference.

Introduction to CFD: Example



Exercise 5.1

Write a second-order solver for the linear advection equation. To mimic a real hydrodynamics code, your code should have routines for finding initializing the state, filling boundary conditions, computing the timestep, constructing the interface states, solving the Riemann problem, and doing the update. The problem flow should look like:

- *set initial conditions*
- *main evolution loop—loop until final time reached*
  - *fill boundary conditions*
  - *get timestep (Eq. 4.5)*
  - *compute interface states (Eqs. 5.6 and 5.7)*
  - *solve Riemann problem at all interfaces (Eq. 5.10)*
  - *do conservative update (Eqs. 5.4 and 5.5)*

Use both the top-hat and Gaussian initial conditions and periodic boundary conditions and compare to the first-order method.

Unit 3. Introduction to Computational Fluid Dynamics (CFD)

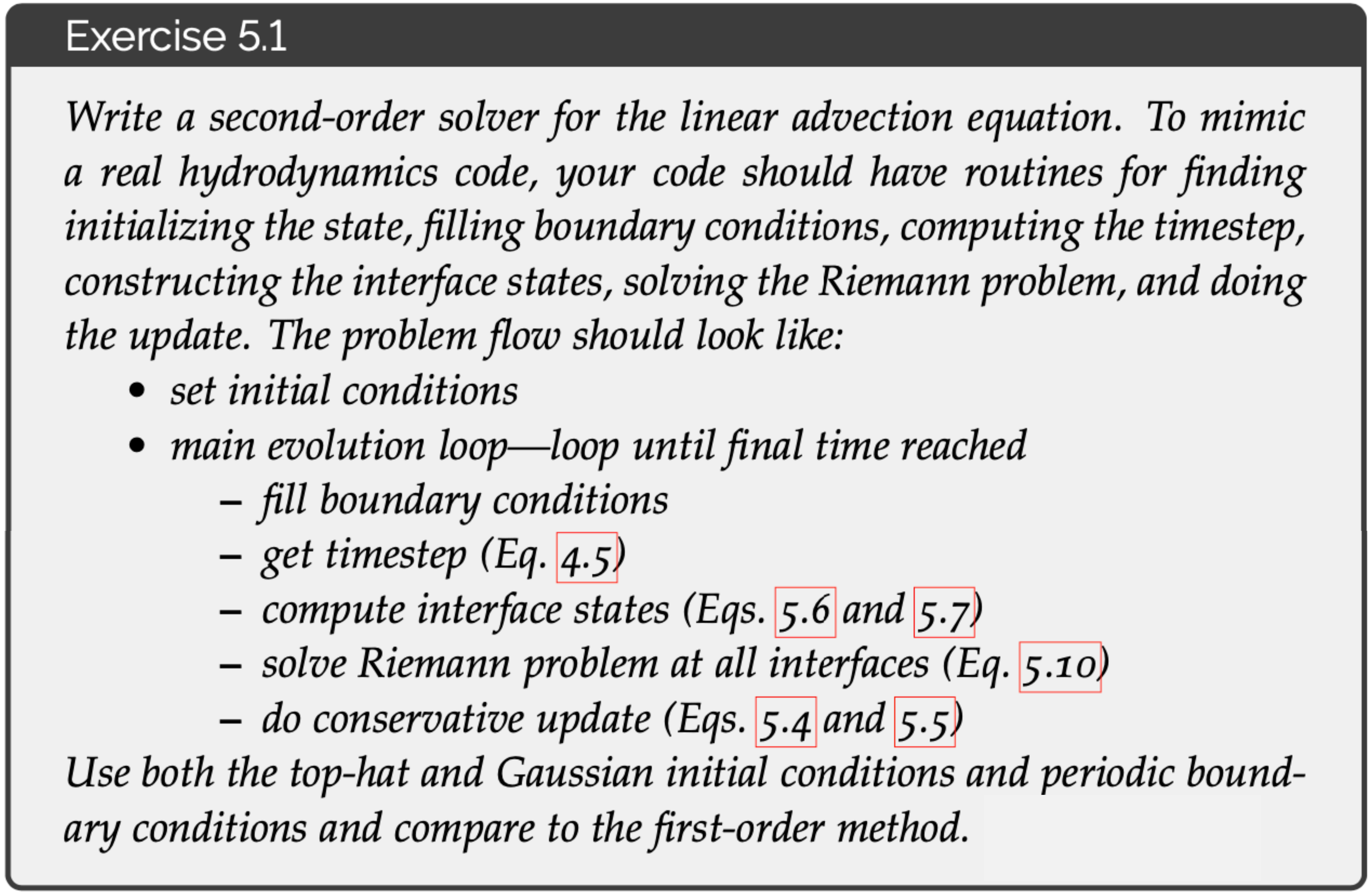
Lecture 315: Slope limiters

Reference book:

“Introduction to Computational Astrophysical Hydrodynamics” by Zingale.
http://bender.astro.sunysb.edu/hydro_by_example/CompHydroTutorial.pdf

W. Banda-Barragán, 2023

Introduction to CFD: Example of finite-volume scheme



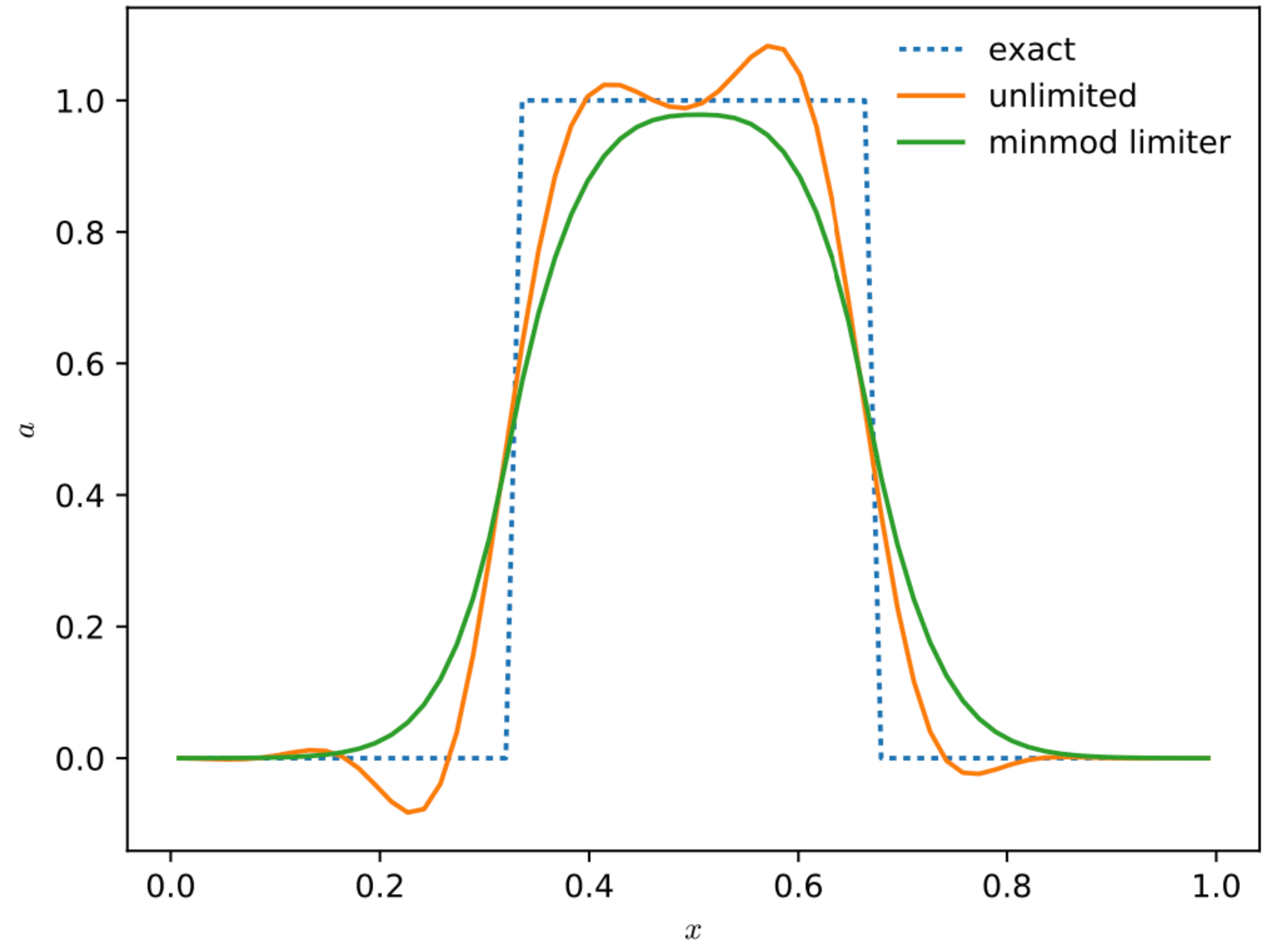
Exercise 5.1

Write a second-order solver for the linear advection equation. To mimic a real hydrodynamics code, your code should have routines for finding initializing the state, filling boundary conditions, computing the timestep, constructing the interface states, solving the Riemann problem, and doing the update. The problem flow should look like:

- *set initial conditions*
- *main evolution loop—loop until final time reached*
  - *fill boundary conditions*
  - *get timestep (Eq. 4.5)*
  - *compute interface states (Eqs. 5.6 and 5.7)*
  - *solve Riemann problem at all interfaces (Eq. 5.10)*
  - *do conservative update (Eqs. 5.4 and 5.5)*

Use both the top-hat and Gaussian initial conditions and periodic boundary conditions and compare to the first-order method.

Introduction to CFD: Limiting



The second-order method likely showed some oscillations in the solution, especially for the top-hat initial conditions.

Godunov's theorem says that any monotonic linear method for advection is first-order accurate.

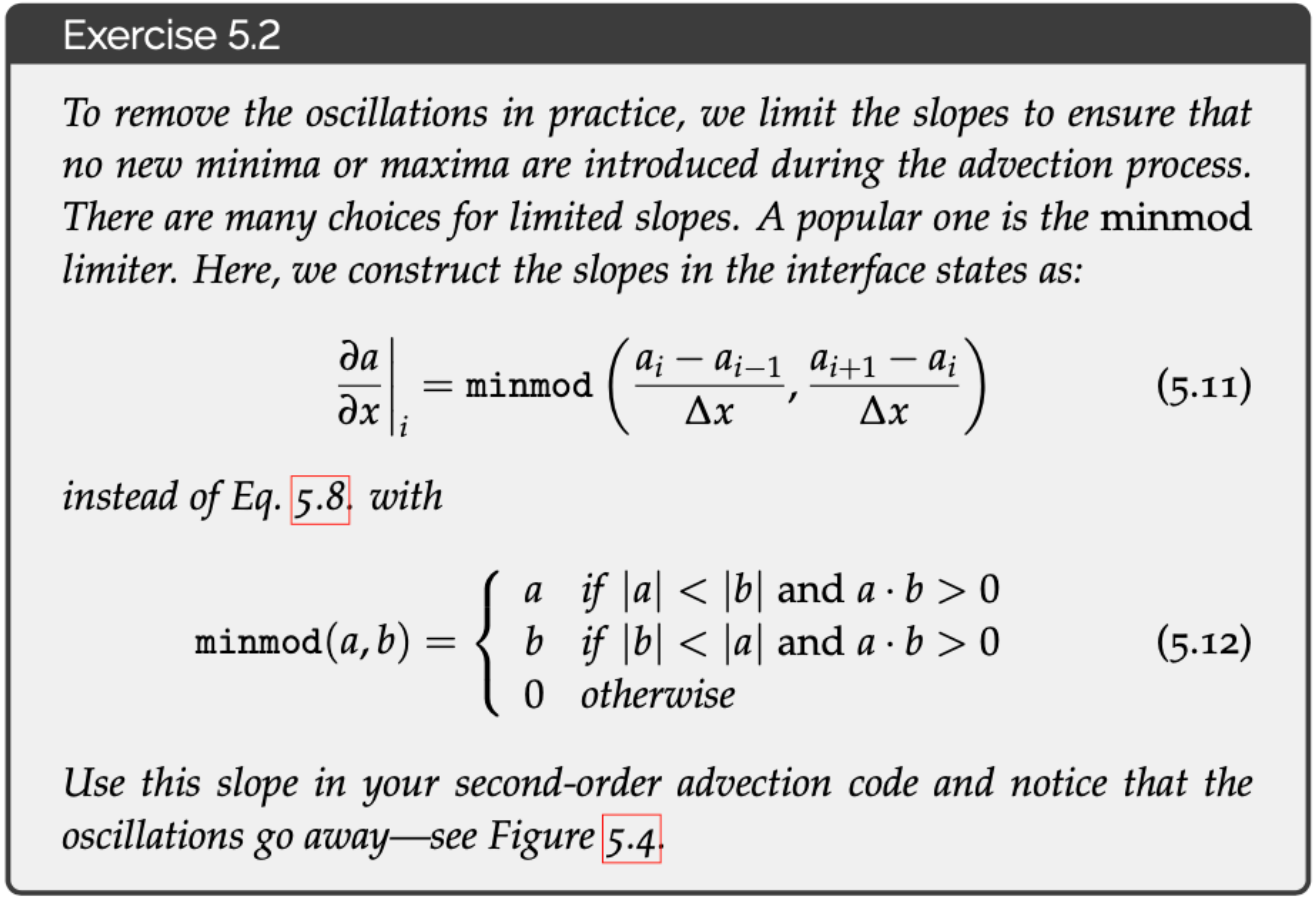
In this context, monotonic means that no new minima or maxima are introduced.

The converse is true too, which suggests that in order to have a second-order accurate method for this linear equation, the algorithm itself must be nonlinear.

Introduction to CFD: the minmod limiter

Slope:

$$\left.\frac{\partial a}{\partial x}\right|_i = \frac{a_{i+1} - a_{i-1}}{2\Delta x} \tag{5.8}$$



Exercise 5.2

To remove the oscillations in practice, we limit the slopes to ensure that no new minima or maxima are introduced during the advection process. There are many choices for limited slopes. A popular one is the minmod *limiter. Here, we construct the slopes in the interface states as:*

$$\left.\frac{\partial a}{\partial x}\right|_i = \texttt{minmod}\left(\frac{a_i - a_{i-1}}{\Delta x}, \frac{a_{i+1} - a_i}{\Delta x}\right) \tag{5.11}$$

instead of Eq. 5.8*. with*

$$\texttt{minmod}(a,b) = \begin{cases} a & \textit{if } |a| < |b| \text{ and } a \cdot b > 0 \\ b & \textit{if } |b| < |a| \text{ and } a \cdot b > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{5.12}$$

Use this slope in your second-order advection code and notice that the oscillations go away—see Figure 5.4*.*

Introduction to CFD: the minmod limiter

We can get a feel for what happens with and without limiting pictorially.

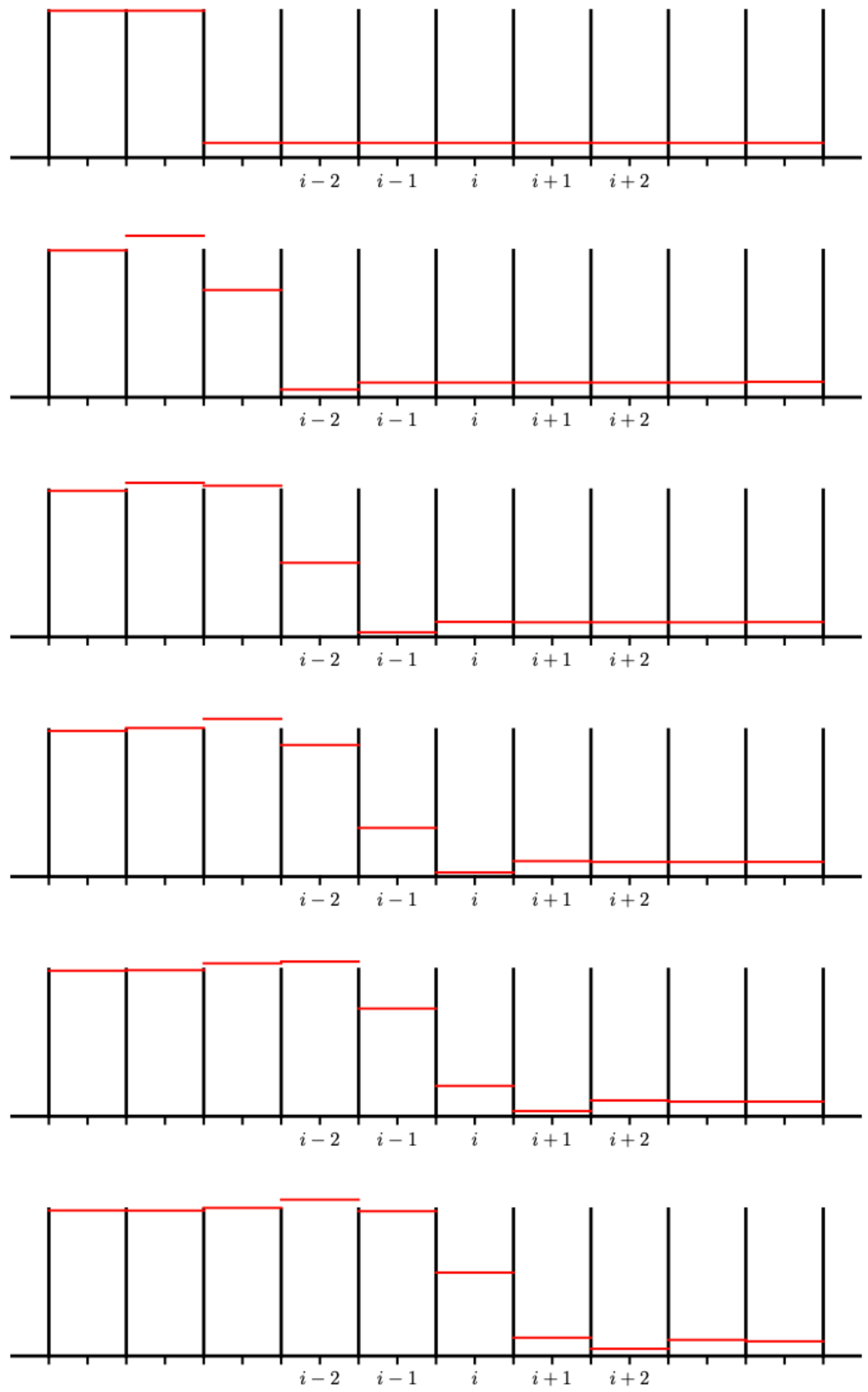
Figures on the next slide show the evolution of an initial discontinuity with and without limiting.

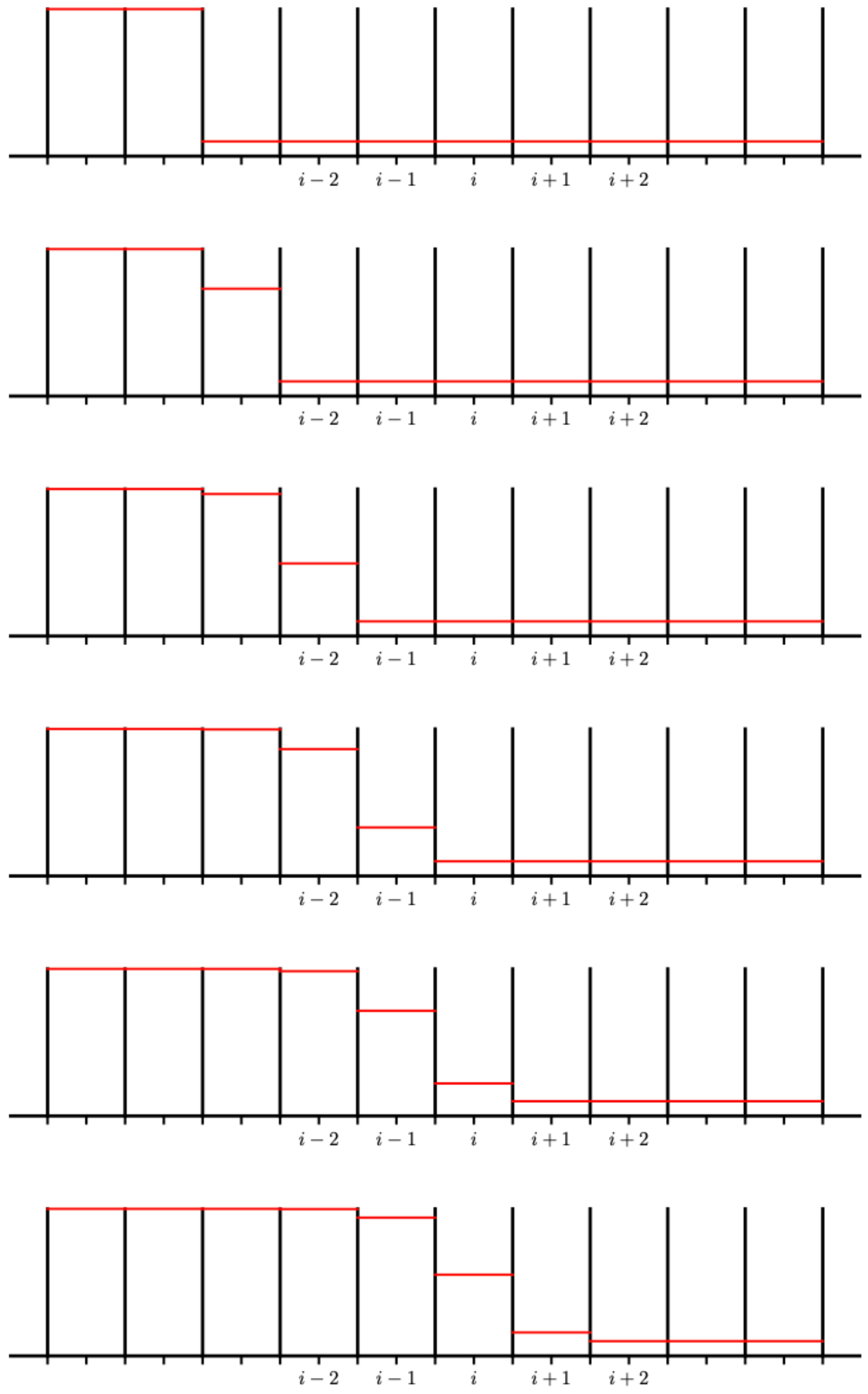
Without limiting, we see an overshoot behind the discontinuity and an undershoot ahead of it—these develop after only a single step.

With each additional step, the overshoots and undershoots move further away from the discontinuity.

In contrast, the case with limiting shows no over- or undershoots around the initial discontinuity. By the end of the evolution, we see that the profile is much narrower in the limiting case than in the case without limiting.

Introduction to CFD: No Limiting versus Limiting


$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$


$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$
$i-2$ $i-1$ i $i+1$ $i+2$

Introduction to CFD: Other limiters?

Note:
Most limiters will have some sort of test on the product of a left-sided and right-sided difference ($a \cdot b$ above)—this is < 0 at an extremum, which is precisely where we want to limit.

A slightly more complex limiter is the MC limiter (monotonized central difference). First we define an extrema test,

$$\xi = (a_{i+1} - a_i) \cdot (a_i - a_{i-1}) \tag{5.13}$$

Then the limited difference is:

$$\left.\frac{\partial a}{\partial x}\right|_i = \begin{cases} \min\left[\frac{|a_{i+1}-a_{i-1}|}{2\Delta x}, 2\frac{|a_{i+1}-a_i|}{\Delta x}, 2\frac{|a_i-a_{i-1}|}{\Delta x}\right] \mathrm{sign}(a_{i+1} - a_{i-1}) & \xi > 0 \\ 0 & \textit{otherwise} \end{cases} \tag{5.14}$$

Introduction to CFD: The MC limiter

The main goal of a limiter is to reduce the slope near extrema. Figure 5.7 shows a finite-volume grid with the original data, cell-centered slopes, and MC limited slopes.

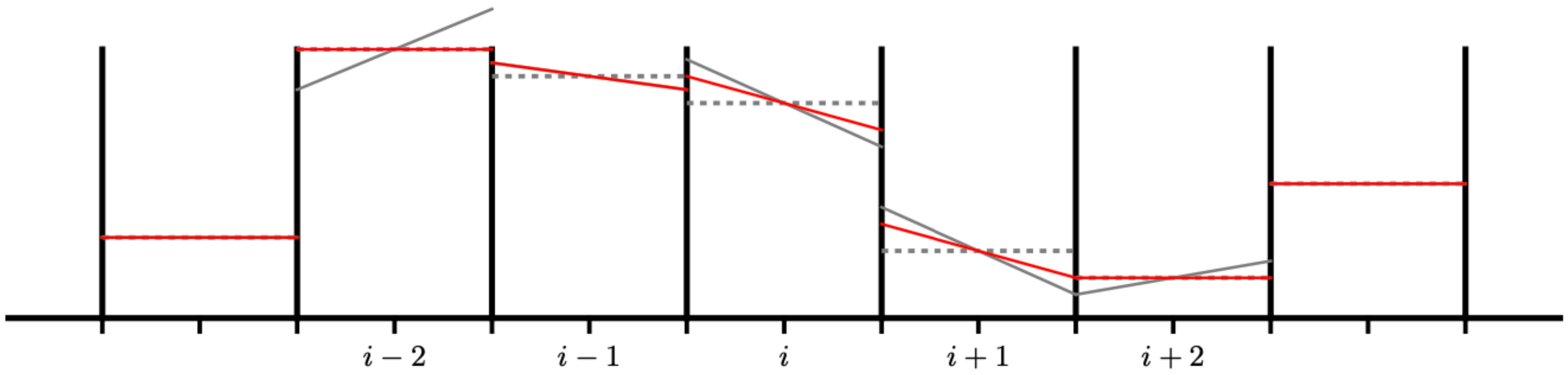


Figure 5.7: A finite-volume grid showing the cell averages (gray, dotted, horizontal lines), unlimited center-difference slopes (gray, solid) and MC limited slopes (red). Note that in zones i and $i+1$, the slopes are limited slightly, so as not to overshoot or undershoot the neighboring cell value. Cell $i-1$ is not limited at all, whereas cells $i-2$, and $i+2$ are fully limited—the slope is set to 0—these are extrema.

Introduction to CFD: The MC limiter

Note that near the strong gradients is where the limiting kicks in. The different limiters are all constructed by enforcing a condition requiring the method to be *total variation diminishing*, or TVD.

It is common to express the slope simply as the change in the state variable:

$$\Delta a_i = \left.\frac{\partial a}{\partial x}\right|_i \Delta x \tag{5.15}$$

and to indicate the limited slope as $\overline{\Delta a_i}$.