On recovering finite-volume methods and studying their properties

Ammar Hakim

March 23rd 2021

1 Exact solution to one-dimensional linear hyperbolic system

Consider the one-dimensional system of equations

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{1}$$

Where $Q(x,t)$ is a vector of m conserved quantities and $F(Q)$ is the flux function. At first, let us assume the system is linear and write the flux as

$$F = AQ \tag{2}$$

where A is a constant $m \times m$ matrix. We will assume that this system is hyperbolic, i.e. the eigenvalues of A are real and the eigenvectors are complete. Let λ_p be the eigenvalues and r^p and l^p, $p = 1, \ldots, m$ be the right and left eigenvectors respectively. We will represent right eigenvectors as column vectors, and left eigenvectors as row vectors.

To solve Eq. (1) we first convert it to a system of uncoupled advection equations by multiplying it from the left by l^p. This gives

$$\frac{\partial w^p}{\partial t} + \lambda_p \frac{\partial w^p}{\partial x} = 0 \tag{3}$$

where we have defined the *Riemann variables* $w^p = l^p Q$. Note that given w^p we can recompute Q from

$$Q = \sum_p w^p r^p. \tag{4}$$

A useful identity is

$$\sum_p w^p \lambda_p r^p = \sum_p w^p A r^p = A \sum_p w^p r^p = AQ = F(Q). \tag{5}$$

Now consider a domain $-\infty < x < \infty$ and the initial conditions $w_0^p(x) = l^p Q_0(x)$. Each advection equation for the Riemann variables can be solved exactly as

$$w^p(x,t) = w_0^p(x - \lambda_p t) \tag{6}$$

From this, the exact solution to the linear system can be obtained as

$$Q(x,t) = \sum_p w_0^p(x - \lambda_p t) r^p = \sum_p l^p Q_0(x - \lambda_p t) r^p. \tag{7}$$

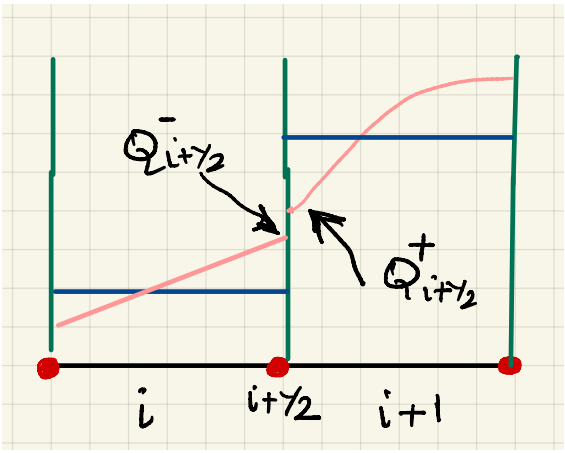

Figure 1: Recovery is used to compute left/right values $Q^{\pm}_{i+1/2}$ from a set of cell-averages and are then fed into a numerical flux function to update the solution.

2 Basic formulation of finite-volume schemes

The finite-volume (FV) scheme for the system Eq. (1) is a method to update the *cell-average* of the solution in time. Consider a cell $I_j \equiv [x_{j-1/2}, x_{j+1/2}]$ with uniform cell-spacing $\Delta x \equiv x_{j+1/2} - x_{j-1/2}$. Integrate Eq. (1) to get

$$\frac{\partial Q_j}{\partial t} + \frac{F_{j+1/2} - F_{j-1/2}}{\Delta x} = 0 \tag{8}$$

where Q_j are cell-average quantities

$$Q_j(t) = \frac{1}{\Delta x} \int_{I_j} Q(x,t)\, dx \tag{9}$$

and $F_{j\pm1/2} = F(Q_{j\pm1/2})$ are the fluxes at the *cell-edges* $x_{j\pm1/2}$. Notice that in effect finite-volume scheme uses the *mean flux gradient* in the cell

$$\frac{1}{\Delta x} \int_{I_j} \frac{\partial F}{\partial x}\, dx = \frac{F_{j+1/2} - F_{j-1/2}}{\Delta x}. \tag{10}$$

to update the *cell average* in time. This is a subtle point and often the cause of misinterpreting the accuracy or order of a FV scheme.

At this point the discrete expression is exact, but only formally: given the cell-averages we can't uniquely determine the edge values $Q_{j\pm1/2}$ to insert into the edge fluxes. The FV scheme is an approximation in which these edge values are *recovered* (approximately) from the cell-averages and then used in a *numerical-flux* to update the cell-average approximation. We can write this as

$$\frac{\partial Q_j}{\partial t} + \frac{G(Q^+_{j+1/2}, Q^-_{j+1/2}) - G(Q^+_{j-1/2}, Q^-_{j-1/2})}{\Delta x} = 0 \tag{11}$$

where $G(Q_L, Q_R)$ the *numerical-flux* function and $Q^+_{j\pm1/2}$ and $Q^-_{j\pm1/2}$ are recovered values just the right and left of the interface $j \pm 1/2$ respectively (see Fig. 1). For consistency with the exact form Eq. (8) we

must ensure that the numeric flux is Lipschitz continuous[1] and *consistent*: when $Q_L = Q_R$ then $G(Q_L, Q_R)$ must reduce to the physical flux function, i.e.

$$\lim_{Q_{L,R} \to Q} G(Q_L, Q_R) = F(Q). \tag{12}$$

Hence, to completely specify a finite-volume scheme we must design algorithms for each of the following three steps:

- **Step 1**: A recovery scheme (possibly with limiters) to compute the left/right interface values $Q_{L,R}$ at each interface using a set of cell-average values around that interface,
- **Step 2**: A numerical flux function that takes the left/right values and returns a consistent approximation to the physical flux, and
- **Step 3**: A time-stepping scheme to advance the solution in time and compute the cell-averages at the next time-step.

3 First-order upwind scheme for one-dimension linear hyperbolic systems

Let us first consider a linear system in which we use the cell-averages directly as the left/right edge values, skipping the recovery step completely. This *first-order* scheme for linear equations will form the basis for higher-order schemes for nonlinear systems. We will also use a simple forward-Euler time-stepper to simplify the third-step.

Instead of directly discretizing the coupled system of equations (in which upwind direction may not not be clear), we can instead solve the linear advection equations for the Riemann variables using an upwind method and the convert the solution for w^p to Q using Eq. (4). We can write a first-order upwind method for the Riemann variables as

$$w_j^{p,n+1} = w_j^{p,n} - \frac{\Delta t}{\Delta x}\left(G^p(w_{j+1}^{p,n}, w_j^{p,n}) - G^p(w_j^{p,n}, w_{j-1}^{p,n})\right) \tag{13}$$

where the numerical flux function for the Riemann variables, $G^p(w_R, w_L)$, is defined as

$$G^p(w_R^p, w_L^p) = \frac{\lambda_p}{2}(w_R^p + w_L^p) - \frac{|\lambda_p|}{2}(w_R^p - w_L^p). \tag{14}$$

Note that this choice of flux ensures that if $\lambda_p > 0$ then $G^p(w_R, w_L) = \lambda_p w_L$ and if $\lambda_p < 0$ then $G^p(w_R, w_L) = \lambda_p w_R$, ensuring proper upwinding of the values at cell interfaces.

To convert this to a scheme for Q instead, multiply by r^p and sum over p to get

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}\left(G(Q_{j+1}^n, Q_j^n) - G(Q_j^n, Q_{j-1}^n)\right). \tag{15}$$

The numerical flux $G(Q_R, Q_L)$ is computed from

$$G(Q_R, Q_L) = \sum_p r^p G^p(w_R^p, w_L^p) = \frac{1}{2}\sum_p \lambda_p r^p (w_R^p + w_L^p) - \frac{1}{2}\sum_p |\lambda_p| r^p (w_R^p - w_L^p). \tag{16}$$

[1] This is somewhat of a technical restriction which ensures that the derivative of the numerical-flux with each of its independent variables is *bounded*.

We can write the first term, using identity Eq. (5) as $(F(Q_R) + F(Q_L))/2$. To rewrite the second term introduce

$$\lambda_p^+ = \max(\lambda_p, 0) \tag{17}$$

$$\lambda_p^- = \min(\lambda_p, 0). \tag{18}$$

Note that in terms of these we can write $\lambda_p = \lambda_p^+ + \lambda_p^-$ and $|\lambda_p| = \lambda_p^+ - \lambda_p^-$. Using the latter identity the numerical flux can be written as

$$G(Q_R, Q_L) = \frac{1}{2}\big(F(Q_R) + F(Q_L)\big) - \frac{1}{2}(A^+\Delta Q_{R,L} - A^-\Delta Q_{R,L}) \tag{19}$$

where the *fluctuations* $A^\pm \Delta Q$ are defined as

$$A^\pm \Delta Q_{R,L} \equiv \sum_p r^p \lambda_p^\pm (w_R^p - w_L^p) = \sum_p r^p \lambda_p^\pm l^p (Q_R - Q_L). \tag{20}$$

The fluctuations satisfy the *flux-difference* or *flux-jump* identity

$$A^+\Delta Q_{R,L} + A^-\Delta Q_{R,L} = \sum_p r^p \underbrace{(\lambda_p^+ + \lambda_p^-)}_{\lambda^p}(w_R^p - w_L^p) = F(Q_R) - F(Q_L). \tag{21}$$

Using this identity, the numerical flux can also be written as

$$G(Q_R, Q_L) = F(Q_L) + A^-\Delta Q_{R,L} = F(Q_R) - A^+\Delta Q_{R,L}. \tag{22}$$

Using this final identity, the complete first-order update for the linear system can be written entirely in terms of fluctuations as

$$Q_j^{n+1} = Q_j^n - \frac{\Delta t}{\Delta x}\left(A^-\Delta Q_{j+1/2} + A^+\Delta Q_{j-1/2}\right). \tag{23}$$

Note that instead, dividing by Δt and taking limits as $\Delta t \to 0$, this can be written in the *semi-discrete* or method-of-lines form

$$\frac{\partial Q_j}{\partial t} = -\frac{1}{\Delta x}\left(A^-\Delta Q_{j+1/2} + A^+\Delta Q_{j-1/2}\right). \tag{24}$$

This system of ODEs can be solved using, for example, a SSP-RK stepper (See Appendix). Further if the left/right edge values $Q_{L,R}$ used in the fluctuations are recovered (and not merely the cell-average values) then we will get a spatially high-order scheme.

4 Fourth-order centered scheme

To construct a high-order scheme we need to recover the interface values $Q_{j\pm1/2}^{+,-}$ from neighboring cell averages. From now on we will drop the indices and use the index-free notation outlined in the Appendix and focus on recovering edge values using four-cells: two on the left and two on the right. At first we will also not worry about limiters.

First, consider the four cells $\{d_{2m}, d_m, d_p, d_{2p}\}$ (two the left of the interface and two to the right). Given the four cell averages each of these four cells we can construct a cubic polynomial

$$p(x) = p_0 + p_1 x + p_2 x^2 + p_3 x^3 \tag{25}$$

which we construct by matching its cell average in cells $\{d_{2m}, d_m, d_p, d_{2p}\}$ to the known cell-average values $\{d_{2m}Q, d_mQ, d_pQ, d_{2p}Q\}$. This gives a system of four equations for the four coefficients p_i, $i = 0, \ldots, 3$ that yield

$$p_0 = \frac{1}{12}(-d_{2m} + 7d_m + 7d_p - d_{2p})Q \tag{26}$$

$$p_1 = \frac{1}{12\Delta x}(d_{2m} - 15d_m + 15d_p - d_{2p})Q \tag{27}$$

$$p_2 = \frac{1}{4\Delta x^2}(d_{2m} - d_m - d_p + d_{2p})Q \tag{28}$$

$$p_4 = \frac{1}{6\Delta x^3}(d_{2m} - 3d_m + 3d_p - d_{2p})Q. \tag{29}$$

Note that the stencils of the even coefficients are *symmetric* and the odd coefficients are *anti-symmetric*. To compute the interface value we do not really need all of these coefficients but only need to evaluate the recovery polynomial at $x = 0$, i.e we only need $p(0) = p_0$. Hence, for this *symmetric* recovery we have the interface values

$$Q^+ = Q^- = \frac{1}{12}(-d_{2m} + 7d_m + 7d_p - d_{2p})Q. \tag{30}$$

In this case of four-cell symmetric recovery the consistency condition shows that the numerical flux function is simply the physical flux. As we will show this leads to a fourth-order *non-dissipative* scheme, however, it is not very robust.

To understand the accuracy of the scheme, consider the advection equation with $F(Q) = Q$. Plugging in the interface values computed using the above recovery formula at the left and right interface gives the complete semi-discrete *five-cell stencil* update formula

$$\frac{\partial Q_j}{\partial t} = -\frac{1}{\Delta x}\int_{I_j}\frac{\partial Q}{\partial x}\,dx = -\frac{Q_{j+1/2} - Q_{j-1/2}}{\Delta x} = -\frac{1}{12\Delta x}(\Delta_{2m} - 8\Delta_m + 8\Delta_p - \Delta_{2p})Q_j \tag{31}$$

where again we have used the index free notation described in the Appendix to express the stencil. A general procedure to compute the accuracy of the scheme is the following

- Take a Taylor series polynomial around the cell center of cell $I_j = [-\Delta x/2, \Delta x/2]$ locally at $x = 0$

$$T(x) = \sum_{n=0}\frac{T_n}{n!}x^n. \tag{32}$$

- Compute the cell average of this polynomial on each of the stencil cells, in this case $\{\Delta_{2m}, \Delta_m, \Delta_p, \Delta_{2p}\}$
- Substitute these averages in the update formula Eq. (31) to compute the mean value of the flux gradient in the cell $I_j = [-\Delta x/2, \Delta x/2]$

$$\frac{1}{12\Delta x}(\Delta_{2m} - 8\Delta_m + 8\Delta_p - \Delta_{2p})T = T_1 + \frac{\Delta x^2}{24}T_3 - \frac{21\Delta x^4}{640}T_5 + \ldots \tag{33}$$

- Subtract the exact cell average of the gradient of the Taylor polynomial in cell $I_j = [-\Delta x/2, \Delta x/2]$, i.e.

$$\frac{1}{\Delta x}\int_{-\Delta x/2}^{\Delta x/2}\frac{\partial T}{\partial x}\,dx = T_1 + \frac{\Delta x^2}{24}T_3 + \frac{\Delta x^4}{1920}T_5 + \ldots \tag{34}$$

  from the stencil computed value. The remainder term is the error of the scheme.

These steps are facilitated by a computer algebra system. For the five-cell stencil above this procedure shows that the error is

$$\frac{\Delta x^4}{30} T_5 + O(\Delta x^6) \tag{35}$$

showing the scheme converges with *fourth-order* accuracy $O(\Delta x^4)$ for linear advection equation.

Besides the accuracy of the scheme we can also study its *diffusion and dispersion* properties by looking at how a single discrete Fourier mode propagates with the scheme. To do this, consider a single mode as

$$Q(x) = e^{ikx} \tag{36}$$

where k is the wavenumber. We will study how this mode is represented by the discrete scheme in the same way as the Taylor series analysis above: first we will compute the cell-average of the mode on each of the cells in the stencil:

$$Q_j = \frac{1}{\Delta x} \int_{x_j - \Delta x/2}^{x_j + \Delta x/2} Q(x)\, dx = \frac{1}{2} \int_{-1}^{1} Q\big(x(\eta)\big)\, d\eta = \frac{1}{2} e^{ikx_j} \int_{-1}^{1} e^{ik\eta \Delta x/2}\, d\eta \tag{37}$$

where we have defined $\eta \equiv 2(x - x_j)/\Delta x$ and x_j is the cell-center coordinate. Now, we can write the generic stencil from different schemes for computing the mean gradient as

$$\frac{1}{\Delta x} \int_{I_j} \frac{\partial Q}{\partial x}\, dx = \frac{1}{\Delta x} \sum_{m=-N}^{M} c_m Q_{j+m}. \tag{38}$$

Plugging in Eq. (37) and Eq. (36) into this expression we get the *effective wavenumber* of the scheme

$$i\overline{k} = \sum_{m=-N}^{M} c_m e^{ikm\Delta x}. \tag{39}$$

Notice we have used $\overline{k}$ instead of k as the wavenumber of the *discrete* mean gradient computed by the FV scheme will not be the same as the wavenumber of the mode initialized on the grid. This difference between the k and $\overline{k}$ is in effect the numerical *dispersion relation* of the scheme. In particular, for a hyperbolic equation linear dispersion relation is of the form $\omega^p = \lambda^p k$ where λ^p are the eigenvalues of the flux-Jacobian. The *semi-discrete scheme*[2] however, will instead have the effective dispersion relation $\omega^p = \lambda^p \overline{k}$. Ideally, one would want $\overline{k} = k$ but this is impossible in the discrete scheme. In fact, all waves in a linear(ized) hyperbolic system should travel with the same constant phase- and group-velocities, but in the discrete system this will *no longer be true*.

Clearly, the discrete wavenumber $\overline{k}$ plays an important role in understanding the properties of the scheme. We would like as broad a range of wavenumbers as possible represented accurately. In general, long wavelength modes will be better represented than short wavelength modes as we will show below. Also, notice from Eq. (39) that $\overline{k}$ could potentially be *complex* even though k is real. Clearly, for stability the imaginary part of $\overline{k}$ must be *negative* or else the solution will blow up in time.

For the five-cell stencil Eq. (31) as the coefficients satisfy $c_m = -c_{-m}$ the discrete wavenumber is *real*, that is it has no numerical diffusion. For long wavelength modes this is perfectly fine, but for shorter wavelength modes the numerical *dispersion* from the discretization won't be damped out, leading to numerical

[2] Additional errors will be introduced with the time-discretization which we will ignore for the present.

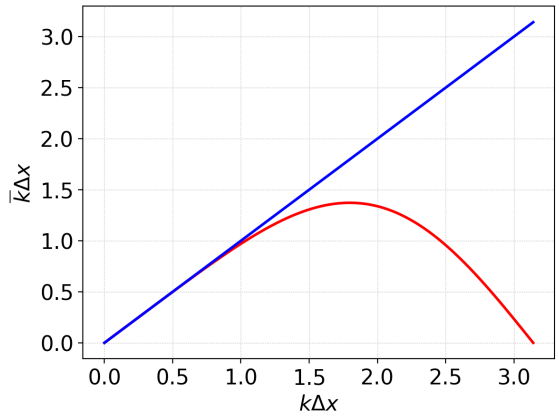

Figure 2: Numerical dispersion relation (red) for the five-cell stencil Eq. (31). For low-k modes (long wavelength) the numerical wavenumber is close to the actual wavenumber (blue) but there is significant dispersion for higher-k modes.

issues whenever sharp gradients are formed in the solution. Note that for hyperbolic problems, in general, such sharp gradients will almost *always* form even if the initial conditions are smooth. For the five-cell stencil we can derive

$$\bar{k}\Delta x = \frac{4}{3}\sin(k\Delta x) - \frac{1}{6}\sin(2k\Delta x). \tag{40}$$

A plot of this numerical dispersion relation is shown in Fig. 2. From this figure it is seen that the numerical wavenumber matches the actual wavenumber for $k\Delta x < 1$ (i.e. a mode needs to be represented by at least six-cells to propagate correctly) but shows significant errors for higher-k modes. In fact, this numerical dispersion relation shows that higher-k modes will suffer significant dispersion, in that the group- and phase-velocities will differ considerably for short wavelength modes. Note that this issue is particularly problematic for turbulence calculations in which the energy cascades down to higher-k modes just where the numerics become problematic. Typically, using an even higher-order (wider) recovery stencil will improve the dispersion charactertics of the scheme at a higher computational (and in parallel, communication) cost. Wider stencils also allow *optimizing* the scheme to reduce the dispersion of high-k modes (in exchange for reduced accuracy) as shown later in this document.

A Strong-Stability preserving Runge-Kutta time-steppers

To update the equations in time we use a strong-stability preserving Runge-Kutta (SSP-RK) scheme. These schemes are ensure that if the basic *first-order* forward Euler time-stepper maintains monotonicity or positivity so does the higher-order RK scheme. To update the equations we first construct a method to take a single first-order Euler step as follows:

$$\mathcal{F}[Q,t] = Q + \Delta t\mathcal{L}[Q,t] \tag{41}$$

Where $\mathcal{L}[Q,t]$ is the RHS of the equation (discretized FV scheme). In terms of this, the most popular scheme is the third-order SSP-RK scheme written as follows:

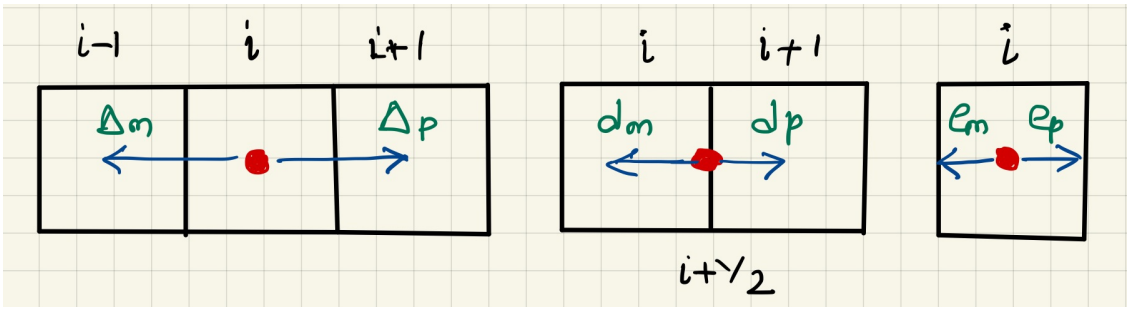

Figure 3: Basic indexing operators to move from cell to cell, face to cell and cell to face. These can be combined with each other, with transverse operators and directional modifiers to express stencils in a compact and dimensionally independent manner.

$$Q^{(1)} = \mathcal{F}[Q^n, t^n] \tag{42}$$

$$Q^{(2)} = \frac{3}{4}Q^n + \frac{1}{4}\mathcal{F}[Q^{(1)}, t^n + \Delta t] \tag{43}$$

$$Q^{n+1} = \frac{1}{3}Q^n + \frac{2}{3}\mathcal{F}[Q^{(2)}, t^n + \Delta t/2] \tag{44}$$

with time-step $\max_p \lambda_p \Delta t/\Delta x \leq 1$.

B Dimensionally independent grid indexing

To allow dimensionally independent grid indexing we will introduce the following operators (see Fig. 3)

- Δ_p and Δ_m work on cell indices and shift to the right and left cell index respectively.
- d_p and d_m work on edge indices and shift it to the right and left full index respectively.
- e_p and e_m work on cell indices and shift it to the right and left edge index respectively.