THE MUSCL-HANCOCK SCHEME FOR SOLUTION OF HYPERBOLIC EQUATIONS

AMMAR H. HAKIM

In this document I outline the MUSCL-Hancock scheme for the solution of 1D hyperbolic partial differential equations. This scheme is a predictor-corrector scheme and is second order accurate in both space and time. We start from the system of hyperbolic 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 $F = F(Q)$ are fluxes. In the following we denote the flux Jacobian as $A \equiv \partial F/\partial Q$ and the eigenvalues as λ^p and the right- and left-eigenvectors as r^p (column vector) and l^p (row vector), for $p = 1, \ldots, m$, respectively.

We will also assume (though this is not required) that the system can be put into the non-conservative ("primitive") quasi-linear form

$$\frac{\partial V}{\partial t} + A_p \frac{\partial V}{\partial x} = 0 \tag{2}$$

where $V(x,t)$ are a vector of m primitive quantities and $A_p(V)$ is a $m \times m$ matrix. Note that any invertible transform $Q = \varphi(V)$ will transform Eq. (1) into Eq. (2).

1. The basic algorithm

The essential idea of the MUSCL-Hancock scheme is to use cell averages to predict the values of the conserved (or primitive) quantities at cell edges at $t + \Delta t/2$ and then use these predicted values to update the solution to $t + \Delta t$. The steps in the algorithm are as follows.

(1) Given cell averages reconstruct a (possibly limited) linear representation of the variables inside each cell. This can be done for either the conserved variables or the primitive variables. Hence, in each cell we represent the solution as

$$W(x,t) = W_i + \frac{x - x_i}{\Delta x}\delta W_i \tag{3}$$

for $x_{i-1/2} < x < x_{i+1/2}$ and where $x_i \equiv (x_{i+1/2} + x_{i-1/2})/2$, $\Delta x \equiv x_{i+1/2} - x_{i-1/2}$ and δW_i are the reconstructed *slopes*. In Eq. (3) $W(x,t)$ stands for the variables we are reconstructing (either primitive or conserved variables). To determine the slopes we can use an averaging procedure

$$\delta W_i = \text{ave}(W_i - W_{i-1}, W_{i+1} - W_i) \tag{4}$$

where $\text{ave}(a,b)$ is a suitable "averaging" function, applied to each component of the vector. Note that using the standard average $\text{ave}(a,b) = (a+b)/2$ leads to a central-difference computed slope, while $\text{ave}(a,b) = 0$ leads to a zero slope or a first-order representation in each cell. Other forms of the average function can be used to avoid spurious oscillations around discontinuities and prevent the formation of unphysical states. See the next section for more details on the reconstruction and averaging steps.

(2) Use the slopes to predict the solution at half time-step, $\Delta t/2$. If the primitive variable slopes have been determined then use the update formula

$$\tilde{V}_j = V_j - \frac{\Delta t}{2\Delta x} A_p(V_i)\delta V_i \tag{5}$$

If the conserved variable slopes have been determined then use the update formula

$$\tilde{Q}_j = Q_j - \frac{\Delta t}{2\Delta x} A(Q_i)\delta Q_i \tag{6}$$

In these formulas $\tilde{V}_j$ and $\tilde{Q}_j$ denote the predicted values in cell $\mathcal{C}_i$.

(3) Use the predicted solution to compute the predicted values at cell edges. As the solution is assumed to be linear, the edge values are

$$W^+_{i-1/2} = \tilde{W}_i - \delta W_i/2 \tag{7}$$

$$W^-_{i+1/2} = \tilde{W}_i + \delta W_i/2 \tag{8}$$

Note that we are using the predicted solution at $t + \Delta t/2$ but the slopes at t to compute the edge values. This gives the edge values at $t + \Delta t/2$ to $O(\Delta t^2)$.

(4) Use the edge values in a Reimann solver (a numerical flux) to update the conserved variables to time t

$$Q^{n+1}_i = Q^n_i - \frac{\Delta t}{\Delta x}(F_{i+1/2} - F_{i+1/2}) \tag{9}$$

where $F_{i\pm 1/2}$ are the numerical fluxes computed from the predicted edge values:

$$F_{i-1/2} \equiv F(W^-_{i-1/2}, W^+_{i-1/2}). \tag{10}$$

See the last section for details on numerical fluxes that can be used in Eq. (9).

2. Reconstruction and limiting

It is simplest to reconstruct each of the conserved variables or the primitive variables directly. This procedure is called *component* reconstruction and limiting. However, a better approach that results in smoother solutions is to limit the *characteristic* variables instead. In this case the limiting is done after projecting the differences on left eigenvectors of the flux Jacobian. Let $L(Q)$ be the matrix of left eigenvectors arranged as rows and let $R(Q)$ be the matrix of right eigenvectors arranged as columns. Note that $L = R^{-1}$. Then the reconstruction becomes, instead of Eq. (4),

$$\delta W_i = R(Q_i)\ \mathrm{ave}(\Delta^i_{i-1}, \Delta^i_i) \tag{11}$$

where $\Delta^j_i = L(Q_j)(W_{i+1} - W_i)$. If the averaging function is non-linear then even for a linear system of the equations the characteristic limiting and component limiting do not coincide.

There are several possible averaging function one can use (besides the zero and simple-averages). For example, the following choices are all designed to avoid unphysical oscillations around discontinuities

- Minmod limiting

$$\mathrm{ave}(a,b) = \begin{cases} \mathrm{minmod}((a+b)/2, 2a, 2b) & \text{if } ab > 0 \\ 0 & \text{if } ab \le 0 \end{cases} \tag{12}$$

- Supebee limiting

$$\mathrm{ave}(a,b) = \begin{cases} \mathrm{minmod}\left(\mathrm{maxmod}(a,b), \mathrm{minmod}(2a,2b)\right) & \text{if } ab > 0 \\ 0 & \text{if } ab \le 0 \end{cases} \tag{13}$$

- Epsilon limiting

$$\mathrm{ave}(a,b) = \frac{(b^2+\epsilon^2)a + (a^2+\epsilon^2)b}{a^2+b^2+2\epsilon^2} \tag{14}$$

where $\epsilon^2 \sim \Delta x^3$ is a parameter.

In the above expressions the $\text{mimod}(a_0, a_1, \ldots)$ function is defined as

$$\text{minmod}(a_0, a_1, \ldots) = \begin{cases} \min(a_0, a_1, \ldots) & \text{if } a_i > 0, \text{ for all } i = 0, 1, \ldots \\ \max(a_0, a_1, \ldots) & \text{if } a_i < 0, \text{ for all } i = 0, 1, \ldots \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

None of the above reconstructions (except the zero-average) ensures that invariant domains are preserved. Another way to put it is that unless something special is done the scheme may not be positivity preserving. For example, while solving the Euler equations the predicted edge values of density and pressure may become negative, leading to unphysical states. A simple but crude way to fix this is to set slopes of *all* quantities in a cell to zero if any of the values at either cell edge becomes negative. More nuanced methods can also be develop by self-consistently[1] adjusting the slopes just enough to ensure invariant domains are preserved.

3. Numerical fluxes

A wide variety of numerical fluxes can be used to compute the edge fluxes needed in Eq. (9). It is important to use a numerical flux that preserves positivity. This combined with a positivity preserving reconstruction will ensure, under a suitable CFL condition, the positivity of the complete scheme.

The simplest numerical flux to use is the local Lax flux (also called the Rusanov flux). This is given by

$$F(Q^-, Q^+) = \frac{F(Q^-) + F(Q^+)}{2} - c\frac{Q^+ - Q^-}{2} \tag{16}$$

Here $c > 0$ is a parameter given by

$$c = \sup_{Q=Q^-,Q^+} \sup_p |\lambda^p|. \tag{17}$$

In other words, the parameter c is the maximum of the absolute eigenvalues computed from the left and right state. Though diffusive, the Lax flux is the simplest in the sense that it requires the minimum amount of information about the equation system being solved: all one needs (besides the flux function) is an *estimate* of the maximum eigenvalue. Note that any c greater than the one computed by Eq. (17) can be used. More complex numerical flux functions that incorporate more information about the equation system can also be used. These flux functions can reduce diffusion as the cost of greater complexity.

[1]What this means is that if the slopes of density and pressure are adjusted, the complete predicted solution (and hence the edge values) must be recomputed with the new slopes.