Lecture 2

Monday, April 4, 2005

Supplementary Reading: Osher and Fedkiw, Sections 3.3 and 3.5; Leveque, Sections 6.7, 8.3, 10.2, 10.4

For a reference on Newton polynomial interpolation via divided difference tables, see Heath, *Scientific Computing*, Section 7.3.3.

In the previous lecture we started with a PDE and discretized it to construct a numerical method. For convergence, we require that the numerical method be both *consistent* and *stable*, possibly imposing conditions on the time step (e.g., $\triangle t < \alpha \triangle x$) to achieve stability.

In this lecture we look at some higher order accurate discretizations. Our goal in using higher order accurate methods is to improve upon consistency. The *order of accuracy* of the method refers to the order of the local truncation error resulting from the discretization. For example,

$O(\triangle x)$	1st order accurate
$O(\triangle x^2)$	2nd order accurate
$O(\triangle x^3)$	3nd order accurate
$\vdots$	$\vdots$

The local truncation error may also have terms such as $O(\triangle t \triangle x)$.

1 TVD Runge-Kutta

To achieve higher order accuracy in the temporal discretization, one can use *Total Variation Diminishing (TVD) Runge-Kutta (RK)* methods. These methods guarantee that the total variation of the solution does not increase, so that no new extrema are generated. For example, if your solution represents a temperature profile, spurious oscillations caused by the numerical method may trigger a chemical reaction in your simulation. Using a TVD method would ensure that this situation does not occur. A related concept is that of Total Variation Bounded (TVB) methods, where the growth in

the total variation is bounded. Both concepts are related to stability. Below we consider 1st order, 2nd order, and 3rd order TVD RK.

- **1st order**, $O(\triangle t)$ error

  The 1st order TVD RK is identical to forward Euler and 1st order RK. It is given by

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + \vec{V}^n \cdot \nabla \phi^n = 0$$

- **2nd order**, $O(\triangle t^2)$ error

  The 2nd order TVD RK method is also known as 2nd order RK, the midpoint rule, modified Euler, and Heun's predictor-corrector method.

  First, an Euler step is taken to advance the solution to time $t^n + \triangle t$

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + \vec{V}^n \cdot \nabla \phi^n = 0 \tag{1}$$

  followed by a second Euler step to advance the solution to time $t^n + 2\triangle t$

$$\frac{\phi^{n+2} - \phi^{n+1}}{\triangle t} + \vec{V}^{n+1} \cdot \nabla \phi^{n+1} = 0 \tag{2}$$

  followed by an averaging step

$$\phi^{n+1} = \frac{1}{2}\phi^n + \frac{1}{2}\phi^{n+2} \tag{3}$$

  that takes a convex combination of the initial data and the result of two Euler steps. The final averaging step produces the second order accurate TVD (or TVB for HJ ENO and HJ WENO) approximation to ϕ at time $t^n + \triangle t$.

- **3rd order**, $O(\triangle t^3)$ error

  First, an Euler step is taken to advance the solution to time $t^n + \triangle t$

$$\frac{\phi^{n+1} - \phi^n}{\triangle t} + \vec{V}^n \cdot \nabla \phi^n = 0 \tag{4}$$

  followed by a second Euler step to advance the solution to time $t^n + 2\triangle t$

$$\frac{\phi^{n+2} - \phi^{n+1}}{\triangle t} + \vec{V}^{n+1} \cdot \nabla \phi^{n+1} = 0 \tag{5}$$

followed by an averaging step

$$\phi^{n+\frac{1}{2}} = \frac{3}{4}\phi^n + \frac{1}{4}\phi^{n+2} \tag{6}$$

that produces an approximation to ϕ at time $t^n + \frac{1}{2}\triangle t$. Then another Euler step is taken to advance the solution to time $t^n + \frac{3}{2}\triangle t$

$$\frac{\phi^{n+\frac{3}{2}} - \phi^{n+\frac{1}{2}}}{\triangle t} + \vec{V}^{n+\frac{1}{2}} \cdot \nabla\phi^{n+\frac{1}{2}} = 0 \tag{7}$$

followed by a second averaging step

$$\phi^{n+1} = \frac{1}{3}\phi^n + \frac{2}{3}\phi^{n+\frac{3}{2}} \tag{8}$$

that produces a third order accurate approximation to ϕ at time $t^n + \triangle t$. This third order accurate TVD RK method has a stability region that includes part of the imaginary axis. Thus, a stable (although ill-advised) numerical method results from combining third order accurate TVD RK with central differencing for the spatial discretization.

2 Hamilton-Jacobi ENO

Recall that in the first order accurate upwind differencing, we approximate the spatial derivative as follows.

if $u_i > 0$, use $\phi_x^- \approx D^-\phi = \frac{\phi_i - \phi_{i-1}}{\triangle x}$

if $u_i < 0$, use $\phi_x^+ \approx D^+\phi = \frac{\phi_{i+1} - \phi_i}{\triangle x}$

if $u_i = 0$, then $u_i\phi_x = 0$, so do nothing

This scheme can be improved upon by using a more accurate approximation for ϕ_x^- and ϕ_x^+. The velocity, u, is still used to decide whether ϕ_x^- or ϕ_x^+ is used, but the approximations for ϕ_x^- or ϕ_x^+ can be improved significantly.

The simplest way to approximate the spatial derivative is to assume that the function is piecewise linear. This is what the approximations D^+ and D^- do. However, we could also pass a parabola or higher order polynomial through the data points. We can then differentiate the interpolated polynomial to get the derivative.

The general idea behind ENO is to use a higher order accurate polynomial to reconstruct ϕ and then to differentiate it to get the approximation to ϕ_x. Such a polynomial is constructed at each grid point. The key to the

algorithm is to choose the neighboring points for the interpolation so that we are not interpolating across steep gradients.

Below we describe the implementation of the HJ ENO method in detail. We do Newton polynomial interpolation via a divided difference table. Given a set of n data points, $(x_1, \phi_1), (x_2, \phi_2), ..., (x_n, \phi_n)$, the Newton polynomial interpolating these points has the form

$$P_{n-1}(x) = \alpha_0 + \alpha_1(x - x_1) + \ldots + \alpha_{n-1}(x - x_1)\ldots(x - x_{n-1})$$

The coefficients, α_j, are the entries in the divided difference table, which we describe below. We use the notation, D_i^k to represent the k^{th} divided difference at grid point i. For example, $D_i^0 = \phi_i$. The first few levels of the divided difference table are constructed as follows.

$$\begin{array}{ll} 0^{th} & D_i^0\phi = \phi_i \\ 1^{st} & D^1_{i+\frac{1}{2}}\phi = \frac{D^0_{i+1}\phi - D^0_i\phi}{\triangle x} \\ 2^{nd} & D^2_i\phi = \frac{D^1_{i+\frac{1}{2}}\phi - D^1_{i-\frac{1}{2}}\phi}{2\triangle x} \\ 3^{rd} & D^3_{i+\frac{1}{2}}\phi = \frac{D^2_{i+1}\phi - D^2_i\phi}{3\triangle x} \\ \vdots & \vdots \end{array}$$

Note that the zeroth level is defined at grid nodes, the first level is defined midway in between grid nodes, the third level at grid nodes, etc. Also from the above table we can see that $D^1_{i+\frac{1}{2}}\phi = (D^+\phi)_i$ and $D^1_{i-\frac{1}{2}}\phi = (D^-\phi)_i$.

The divided differences are used to reconstruct a polynomial of the form

$$\phi(x) = Q_0(x) + Q_1(x) + Q_2(x) + Q_3(x) \tag{9}$$

that can be differentiated and evaluated at x_i to find $(\phi_x^-)_i$ and $(\phi_x^+)_i$. That is, we use

$$\phi_x(x_i) = Q_1'(x_i) + Q_2'(x_i) + Q_3'(x_i) \tag{10}$$

to define $(\phi_x^-)_i$ and $(\phi_x^+)_i$. Note that the constant $Q_0(x)$ term vanishes upon differentiation.

To find ϕ_x^- we start with $k = i - 1$, and to find ϕ_x^+ we start with $k = i$. Then we define

$$Q_1(x) = (D^1_{k+1/2}\phi)(x - x_i) \tag{11}$$

so that

$$Q_1'(x_i) = D^1_{k+\frac{1}{2}}\phi \tag{12}$$

implying that the contribution from $Q_1'(x_i)$ in equation 10 is the backward difference in the case of ϕ_x^- and the forward difference in the case of ϕ_x^+. In other words, first order accurate polynomial interpolation is exactly first order upwinding. Improvements are obtained by including the $Q_2'(x_i)$ and $Q_3'(x_i)$ terms in equation 10 leading to second and third order accuracy respectively.

Looking at the divided difference table and noting that $D^1_{k+1/2}\phi$ was chosen for first order accuracy, we have two choices for the second order accurate correction. We could include the next point to the left and use $D^2_k\phi$, or we could include the next point to the right and use $D^2_{k+1}\phi$. The key observation is that smooth slowly varying data tends to produce small numbers in divided difference tables while discontinuous or quickly varying data tends to produce large numbers in divided difference tables. This is obvious in the sense that the differences measure variation in the data. Comparing $|D^2_k\phi|$ to $|D^2_{k+1}\phi|$ indicates which of the polynomial interpolants has more variation. We would like to avoid interpolating near large variations such as discontinuities or steep gradients, since they cause overshoots in the interpolating function leading to numerical errors in the approximation of the derivative. Thus, if $|D^2_k\phi| \leq |D^2_{k+1}\phi|$ we set $c = D^2_k\phi$ and $k^\star = k-1$, otherwise we set $c = D^2_{k+1}\phi$ and $k^\star = k$. Then we define

$$Q_2(x) = c(x - x_k)(x - x_{k+1}) \tag{13}$$

so that

$$Q_2'(x_i) = c\,(2(i-k)-1)\,\triangle x \tag{14}$$

is the second order accurate correction to the approximation of ϕ_x in equation 10. If we stop here, i.e. omitting the Q_3 term, we have a second order accurate method for approximating ϕ_x^- and ϕ_x^+. Note that $k^\star$ was not yet used. It is only defined for use below when calculating the third order accurate correction.

Similar to the second order accurate correction, the third order accurate correction is obtained by comparing $|D^3_{k^\star+1/2}\phi|$ and $|D^3_{k^\star+3/2}\phi|$. If $|D^3_{k^\star+1/2}\phi| \leq |D^3_{k^\star+3/2}\phi|$ we set $c^\star = D^3_{k^\star+1/2}\phi$, otherwise we set $c^\star = D^3_{k^\star+3/2}\phi$. Then we define

$$Q_3(x) = c^\star(x - x_{k^\star})(x - x_{k^\star+1})(x - x_{k^\star+2}) \tag{15}$$

so that

$$Q_3'(x_i) = c^\star\left(3(i-k^\star)^2 - 6(i-k^\star) + 2\right)(\triangle x)^2 \tag{16}$$

is the third order accurate correction to the approximation of ϕ_x in equation 10.