Computational Fluid Dynamics
http://www.nd.edu/~gtryggva/CFD-Course/

Advection

Grétar Tryggvason
Spring 2013

Computational Fluid Dynamics

Higher Order and more recent methods

Computational Fluid Dynamics
ENO/WENO

In many cases we have solutions that require a high order method away from the discontinuity to represent a rapidly varying but smooth solution.

Beyond linear: Reconstruction of higher order approximations for the function in each cell (ENO and WENO).

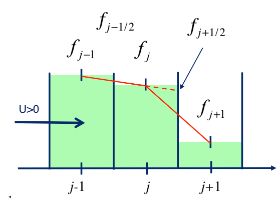
The critical step in the methods discussed so far is the construction of a linear slope in each cell and the limitation of this slope to prevent oscillations. For higher order methods, a higher order profile needs to be constructed

---

Computational Fluid Dynamics
ENO/WENO

Example: Second order ENO

1. Construct left and right slopes by connecting the average values in adjacent cells
2. Select the downstream flux by using the smaller slope



$$\Delta f_j^+ = f_{j+1} - f_j \qquad \Delta f_j^- = f_j - f_{j-1}$$

$$f_{j+1/2} = \begin{cases} f_j + \frac{1}{2}\operatorname{amin}\left(\Delta f_j^+, \Delta f_j^-\right), & \text{if } \frac{1}{2}\left(u_j + u_{j+1}\right) > 0 \\ f_j - \frac{1}{2}\operatorname{amin}\left(\Delta f_{j+1}^+, \Delta f_{j+1}^-\right), & \text{if } \frac{1}{2}\left(u_j + u_{j+1}\right) < 0 \end{cases}$$

Computational Fluid Dynamics
ENO/WENO

Second order ENO scheme for the linear advection equation

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0 \qquad \operatorname{amin}(a,b) = \begin{cases} a, & |a| < |b| \\ b, & |b| \le |a| \end{cases}$$

$$f_j^* = f_j^n - \frac{\Delta t}{h} u_j^n \left(f_{j+1/2}^n - f_{j-1/2}^n\right)$$

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{h}\frac{1}{2}\left(u_j^n\left(f_{j+1/2}^n - f_{j-1/2}^n\right) + u_j^*\left(f_{j+1/2}^* - f_{j-1/2}^*\right)\right)$$

$$f_{j+1/2} = \begin{cases} f_j + \frac{1}{2}\operatorname{amin}\left(\Delta f_j^+, \Delta f_j^-\right), & \text{if } \frac{1}{2}\left(u_j + u_{j+1}\right) > 0 \\ f_j - \frac{1}{2}\operatorname{amin}\left(\Delta f_{j+1}^+, \Delta f_{j+1}^-\right), & \text{if } \frac{1}{2}\left(u_j + u_{j+1}\right) < 0 \end{cases}$$

$$\Delta f_j^+ = f_{j+1} - f_j \qquad \Delta f_j^- = f_j - f_{j-1}$$

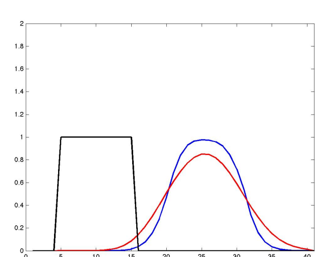
Computational Fluid Dynamics
ENO/WENO

Second order ENO scheme for the linear advection equation

$$\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0$$

Blue: 2nd order ENO

Red: 1st Upwind



Computational Fluid Dynamics

Generalize to higher order

Computational Fluid Dynamics

$$\frac{\partial f}{\partial t}+\frac{\partial F}{\partial x}=0 \qquad \bar{f}_i(t)=\frac{1}{\Delta x}\int_{V_i} f(x,t)\,dx$$

$$\frac{d}{dt}\bar{f}_i(t)=\frac{1}{\Delta x}\big(F(f(x_{i+1/2},t)-F(f(x_{i+1/2},t)\big)$$

$$=\frac{1}{\Delta x}\big(F_{i+1/2}-F_{i-1/2}\big)=L(\bar{f})_i$$

For high order methods the time integration is often done using high order Runge-Kutta methods, such the following third order method:

$$\bar{f}^{(1)}=\bar{f}^{n}+\Delta t\,L(\bar{f}^{n})$$

$$\bar{f}^{(2)}=\frac{3}{4}\bar{f}^{n}+\frac{1}{4}\bar{f}^{(1)}+\frac{1}{4}\Delta t\,L(\bar{f}^{(1)})$$

$$\bar{f}^{n+1}=\frac{1}{3}\bar{f}^{n}+\frac{2}{3}\bar{f}^{(2)}+\frac{2}{3}\Delta t\,L(\bar{f}^{(2)})$$

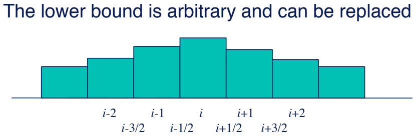
Computational Fluid Dynamics
ENO/WENO

Constructing an interpolation polynomial from the cell averages: For anything higher than second order (linear) the problem is that the average value in the cell is not equal to the value at the center.

To get around this we look at the primitive function:

$$V(x)\equiv\int_{-\infty}^{x} v(\xi)\,d\xi$$

The lower bound is arbitrary and can be replaced

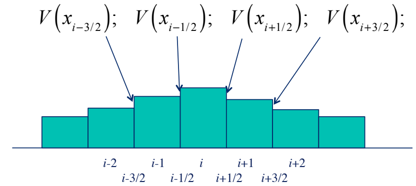


Computational Fluid Dynamics
ENO/WENO

Since this is the integral over the cells, the discrete version is exact at the cell boundaries

$$V\left(x_{i+1/2}\right)=\sum_{j=-\infty}^{i}\int_{x_{j-1/2}}^{x_{j+1/2}} v\left(\xi\right)d\xi=\sum_{j=-\infty}^{i} f_i\Delta x$$

$$V\left(x_{i-3/2}\right);\quad V\left(x_{i-1/2}\right);\quad V\left(x_{i+1/2}\right);\quad V\left(x_{i+3/2}\right);$$



Computational Fluid Dynamics
ENO/WENO

A polynomial interpolating the edge values is given by $P(x)$ and we denote its derivative by $p(x)$

$$p(x)=P'(x)$$

Then it can be shown that

$$\int_{x_{i-1/2}}^{x_{i+1/2}} p\left(\xi\right)d\xi=\int_{x_{i-1/2}}^{x_{i+1/2}} P'\left(\xi\right)d\xi=P(x_{i+1/2})-P(x_{i-1/2})$$

$$=V(x_{i+1/2})-V(x_{i-1/2})=\int_{x_{i-1/2}}^{x_{i+1/2}} v\left(\xi\right)d\xi=f_i\Delta x$$

That is, the integral of $p(x)$ over the cell is equal to the cell average f_i

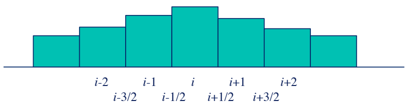
---

Computational Fluid Dynamics
ENO/WENO

Thus, $p(x)$ gives the correct average value in each cell and the integrated value gives the exact values of the primitive function at the cell boundaries.

We need to write down a polynomial $P(x)$ that interpolates the values of the primitive function of the cell boundaries and then differentiate this polynomial to get $p(x)$, which lets us compute the variables at the cell boundary

Computational Fluid Dynamics
ENO/WENO



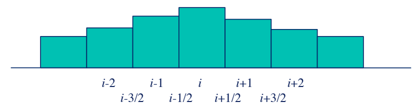
The interpolation polynomial is often taken to be the Lagrangian Polynomial

$$P(x)=\sum_{m=0}^{k} V(x_{i-r+m-1/2})\prod_{\substack{l=0\\ l\neq m}}^{k}\frac{x-x_{i-r+l-1/2}}{x_{i-r-1/2}-x_{i-r+l-1/2}}$$

Where r determines where we start and k is the order

Computational Fluid Dynamics
ENO/WENO

ENO: Essential Non-Oscillatory



The question is now which point we select. We start by interpolating over one cell (linear). To add one point we can add either the point to the left or the right. In ENO we select the points based on the minimum absolute value of the divided differences of the function values

Computational Fluid Dynamics
ENO/WENO

WENO

Weighted Essential Non-Oscillatory

Reference: C-W Shu. High Order Weighted Essential Nonoscillatory Schemes for Convection Dominated Problems. SIAM Review, Vol. 51 (2009), 82-126.

Computational Fluid Dynamics
ENO/WENO

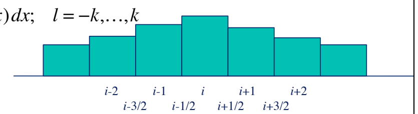
$$\frac{\partial f}{\partial t}+\frac{\partial F}{\partial x}=0 \qquad \bar{f}_i(t)=\frac{1}{\Delta x}\int_{V_i} f(x,t)\,dx$$

$$\frac{d}{dt}\bar{f}_i(t)=\frac{1}{\Delta x}\Big(F(f(x_{i+1/2},t)-F(f(x_{i+1/2},t)\Big)=\frac{1}{\Delta x}\Big(F_{i+1/2}-F_{i-1/2}\Big)$$

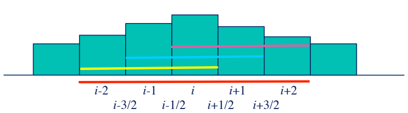
Find the cell average by integrating polynomial representation of the function, first over a subset of the points and then over all the points

$$\bar{f}_{i+l}=\frac{1}{\Delta x}\int_{V_{i+l}} p_j(x)\,dx; \quad l=-k+j,\ldots,j$$

$$\bar{f}_{i+l}=\frac{1}{\Delta x}\int_{V_{i+l}} Q(x)\,dx; \quad l=-k,\ldots,k$$



Computational Fluid Dynamics
ENO/WENO



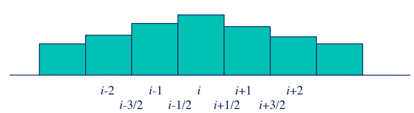
For a polynomial of order three, over the three intervals indicated we get

$$p_0(x_{i+1/2})=\frac{1}{3}\bar{f}_{i-2}-\frac{7}{6}\bar{f}_{i-1}+\frac{11}{6}\bar{f}_i$$

$$p_1(x_{i+1/2})=-\frac{1}{6}\bar{f}_{i-1}+\frac{5}{6}\bar{f}_i+\frac{1}{3}\bar{f}_{i+1}$$

$$p_2(x_{i+1/2})=\frac{1}{3}\bar{f}_i+\frac{5}{6}\bar{f}_{i+1}-\frac{1}{6}\bar{f}_{i+2}$$

Computational Fluid Dynamics
ENO/WENO



A polynomial for the whole interval is given by

$$Q(x_{i+1/2})=\frac{1}{30}\bar{f}_{i-2}-\frac{13}{60}\bar{f}_{i-1}+\frac{47}{60}\bar{f}_i+\frac{0}{20}\bar{f}_{i+1}-\frac{1}{30}\bar{f}_{i+2}$$

Which is a weighted average of the lower order polynomials

$$Q(x_{i+1/2})=\sum_{j=0}^{k}\gamma_j p_j(x_{i+1/2}) \qquad \gamma_0=\frac{1}{10};\quad \gamma_1=\frac{6}{10};\quad \gamma_2=\frac{3}{10};$$

Computational Fluid Dynamics
ENO/WENO

Introduce a smoothness measure

$$\beta_j = \sum_{l=1}^{k} \int_{V_i} \Delta x^{2l-1} \left(\frac{\partial^l}{\partial x^l} p_j(x) \right)^2 dx$$

For our case this gives:

$$\beta_0 = \frac{13}{12}\left(\bar{f}_{i-2} - 2\bar{f}_{i-1} + \bar{f}_i\right)^2 + \frac{1}{4}\left(3\bar{f}_{i-2} - 4\bar{f}_{i-1} + \bar{f}_i\right)^2$$

$$\beta_1 = \frac{13}{12}\left(\bar{f}_{i-1} - 2\bar{f}_i + \bar{f}_{i+1}\right)^2 + \frac{1}{4}\left(3\bar{f}_{i-1} - 4\bar{f}_i + \bar{f}_{i+1}\right)^2$$

$$\beta_2 = \frac{13}{12}\left(\bar{f}_i - 2\bar{f}_{i+1} + \bar{f}_{i+2}\right)^2 + \frac{1}{4}\left(3\bar{f}_i - 4\bar{f}_{i+1} + \bar{f}_{i+2}\right)^2$$

Computational Fluid Dynamics
ENO/WENO

Then compute weights to find the smoothes approximation to the value of *f* at the cell boundary.

First find:

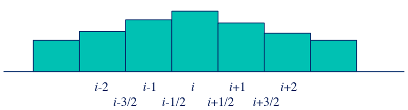
$$\omega_j = \frac{\bar{\omega}_j}{\sum_j \bar{\omega}_j}; \quad \bar{\omega}_j = \frac{\gamma_j}{\sum_j (\varepsilon + \beta_j)}$$

Then compute:

$$f^-_{i+1/2} \approx \sum_{j=0}^{k} \omega_j p_j(x_{i+1/2})$$

The value on the other side is found in the same way

Computational Fluid Dynamics
ENO/WENO



In the WENO (weighted essentially non-oscillating) scheme we use all the points but weigh the contribution of each according to a smoothness criteria. High-order WENO represents the current state-of-the-art in computing of flows with sharp interfaces

Other smoothness criteria, weights, and interpolation functions have been studies, as well as how to implement the method on non-structured grids.

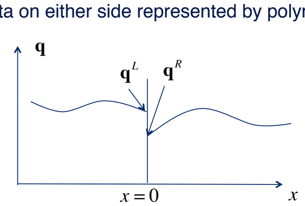
Computational Fluid Dynamics

Other Approaches

Computational Fluid Dynamics

ADER Schemes: **Arbitrary DERivatives**

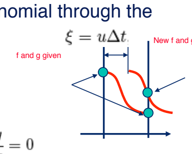
Solve a higher order (generalized) Riemann problem for smooth data on either side represented by polynomials



Computational Fluid Dynamics
CIP-gradient augmentation

The CIP (Constrained Interpolation Polynomial) Method (Yabe)

In addition to advecting the marker function f, its derivative is advected by fitting a third order polynomial through the function and its derivatives.



Start with $\frac{\partial f}{\partial t} + u\frac{\partial f}{\partial x} = 0$

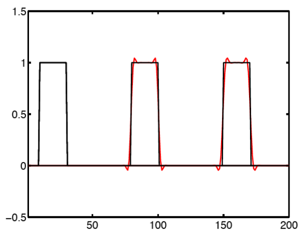
Introduce $g = \partial f / \partial x$.

In 1D, the advection of the derivative is given by $\frac{\partial g}{\partial t} + u\frac{\partial g}{\partial x} = 0$

Therefore, the derivative is translated with velocity u, just as the function. In 2D splitting is used to separate translation and deformation

Computational Fluid Dynamics
CIP-gradient augmentation

The CIP method results in very accurate advection and for a sharp interface it greatly reduces overshoots, but does not eliminate them completely



Computational Fluid Dynamics

Although most high-order finite volume methods are based on updating the average value and reconstruct a higher order approximation, advecting more information is gaining some popularity

---

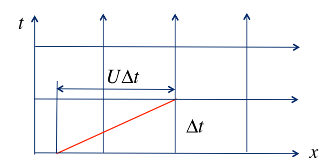
Computational Fluid Dynamics

Semi-Lagrangian Schemes

Computational Fluid Dynamics

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=0$$

$$\frac{df}{dt}=0 \quad \text{on} \quad \frac{dx}{dt}=U \qquad f(t,x)=f(t-\Delta t,x-U\Delta t)$$



For the linear advection equation the solution is advected without change

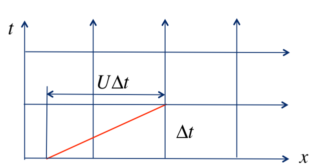
---

Computational Fluid Dynamics

$$f(t,x)=f(t-\Delta t,x-U\Delta t)$$

$$f_j^{n+1}=\mathrm{Intp}(f^n(x_j^o))$$

$$x_j^o=x_j-U\Delta t$$



Interpolate the solution at the old time level, at a point given by tracing back the characteristic

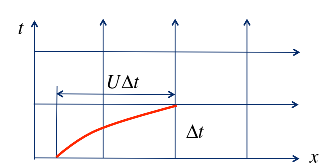
---

Computational Fluid Dynamics

$$\frac{\partial f}{\partial t}+u(x,t)\frac{\partial f}{\partial x}=0 \qquad x_j^o=x_j-\int_t^{t+\Delta t}u(x(t),t)\,dt$$

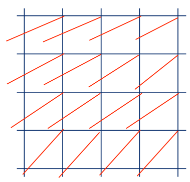
$$x_j^o=x_j-\frac{\Delta t}{2}\left(u(x_j,t)+u(x_j^o,t+\Delta t)\right)$$

$$\frac{df}{dt}=0 \quad \text{on} \quad \frac{dx}{dt}=u \qquad f_j^{n+1}=\mathrm{Intp}(f^n(x_j^o))$$



For higher order methods an iteration is necessary

For two and three-dimensional flows a multidimensional interpolation is necessary



Linear interpolation is usually too diffusive but several higher order ones have been used

Semi-Lagrangian schemes are widely used in weather forecasting and simulations of plasma, for example

Flux Vector Splitting

For upwind schemes, it is necessary to determine the upstream direction. For systems with many characteristics running both left and right, there is not one "upstream" direction

Upwind Scheme - Revisited

Generalized Upwind Scheme (for both $U>0$ and $U<0$)

$$f_j^{n+1} = f_j^n - \frac{U\Delta t}{h}(f_j^n - f_{j-1}^n), \quad U>0$$

$$f_j^{n+1} = f_j^n - \frac{U\Delta t}{h}(f_{j+1}^n - f_j^n), \quad U<0$$

Define

$$U^+ = \frac{1}{2}(U+|U|), \quad U^- = \frac{1}{2}(U-|U|)$$

The two cases can be combined into a single expression:

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{h}\left[U^+(f_j^n - f_{j-1}^n) + U^-(f_{j+1}^n - f_j^n)\right]$$

Where we have split the flux in an "upwind" and "downwind" part

For a system of equations there are generally waves running in both directions. To apply upwinding, the fluxes must be decomposed into left and right running waves

Computational Fluid Dynamics
Flux Splitting

A system of hyperbolic equations

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}}{\partial x}=0$$

Steger-Warming (1979)

can be written in the form

$$\frac{\partial \mathbf{f}}{\partial t}+[A]\frac{\partial \mathbf{f}}{\partial x}=0; \qquad [A]=\frac{\partial \mathbf{F}}{\partial \mathbf{f}}$$

The system is hyperbolic if

$$[T]^{-1}[A][T]=[\lambda]; \quad [T]^{-1}=\begin{bmatrix}\mathbf{q}_1^{\mathrm{T}}\\ \vdots \\ \mathbf{q}_N^{\mathrm{T}}\end{bmatrix}$$

Computational Fluid Dynamics
Flux Splitting

$$\mathbf{F}=[A]\mathbf{f}=[T][\lambda][T]^{-1}\mathbf{f}$$

The matrix of eigenvalues $[\lambda]$ is divided into two matrices

$$[\lambda]=[\lambda^{+}]+[\lambda^{-}]$$

Hence $[A]=[A^{+}]+[A^{-}]=[T][\lambda^{+}][T]^{-1}+[T][\lambda^{-}][T]^{-1}$

Define $\mathbf{F}=\mathbf{F}^{+}+\mathbf{F}^{-}$ $\quad \left(\mathbf{F}^{+}=[A^{+}]\mathbf{f},\ \ \mathbf{F}^{-}=[A^{-}]\mathbf{f}\right)$

Conservation law becomes

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}=0$$

Computational Fluid Dynamics
Flux Splitting

Example: 1-D Hyperbolic Equation

$$\frac{\partial^2 f}{\partial t^2}-c^2\frac{\partial^2 f}{\partial x^2}=0$$

Leading to

$$\begin{pmatrix}v_t\\ w_t\end{pmatrix}+\begin{pmatrix}0 & -c^2\\ -1 & 0\end{pmatrix}\begin{pmatrix}v_x\\ w_x\end{pmatrix}=0 \qquad v=\frac{\partial f}{\partial t};\quad w=\frac{\partial f}{\partial x}$$

$$\mathbf{f}=\begin{pmatrix}v\\ w\end{pmatrix}; \quad \mathbf{F}=\begin{pmatrix}-c^2 w\\ -v\end{pmatrix}; \quad [A]=\left[\frac{\partial \mathbf{F}}{\partial \mathbf{f}}\right]=\begin{bmatrix}0 & -c^2\\ -1 & 0\end{bmatrix}$$

Computational Fluid Dynamics
Flux Splitting

$$[T]^{-1}=\begin{bmatrix}\mathbf{q}_1^{\mathrm{T}}\\ \mathbf{q}_2^{\mathrm{T}}\end{bmatrix}=\begin{bmatrix}1 & -c\\ 1 & c\end{bmatrix} \qquad [\lambda^{+}]=\begin{vmatrix}c & 0\\ 0 & 0\end{vmatrix}; \quad [\lambda^{-}]=\begin{vmatrix}0 & 0\\ 0 & -c\end{vmatrix}$$

$$[A^{+}]=[T][\lambda^{+}][T]^{-1}; \quad [A^{-}]=[T][\lambda^{-}][T]^{-1}$$

$$\mathbf{F}^{+}=[A^{+}]\mathbf{f}; \quad \mathbf{F}^{-}=[A^{-}]\mathbf{f}$$

Leading to: $\mathbf{F}^{+}=\dfrac{1}{2}\begin{bmatrix}cv-c^2 w\\ -v+cw\end{bmatrix}; \quad \mathbf{F}^{-}=\dfrac{1}{2}\begin{bmatrix}-cv-c^2 w\\ -v-cw\end{bmatrix}$

For a nonlinear system of equations such as the Euler equations, there is some arbitrariness in how the system is split

Computational Fluid Dynamics

Solve using first order upwinding with flux splitting

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}=0 \qquad \mathbf{F}^{+}=\frac{1}{2}\begin{bmatrix}cv-c^2 w\\ -v+cw\end{bmatrix}; \quad \mathbf{F}^{-}=\frac{1}{2}\begin{bmatrix}-cv-c^2 w\\ -v-cw\end{bmatrix}$$

$$\begin{pmatrix}v_t\\ w_t\end{pmatrix}+\begin{pmatrix}0 & -c^2\\ -1 & 0\end{pmatrix}\begin{pmatrix}v_x\\ w_x\end{pmatrix}=0 \qquad v=\frac{\partial f}{\partial t};\quad w=\frac{\partial f}{\partial x}$$

Finite volume upwind approximation:

$$\begin{bmatrix}v\\ w\end{bmatrix}_j^{n+1}=\begin{bmatrix}v\\ w\end{bmatrix}_j^{n}-\frac{\Delta t}{h}\frac{1}{2}\left(\begin{bmatrix}cv-c^2w\\ -v+cw\end{bmatrix}_j^n-\begin{bmatrix}cv-c^2w\\ -v+cw\end{bmatrix}_{j-1}^n+\begin{bmatrix}-cv-c^2w\\ -v-cw\end{bmatrix}_{j+1}^n-\begin{bmatrix}-cv-c^2w\\ -v-cw\end{bmatrix}_j^n\right)$$

Computational Fluid Dynamics

Enormous progress has been made in solution techniques for hyperbolic systems with shocks in the last twenty years. Advanced methods are now able to resolve complex shocks within a grid space or two, even in multidimensional situations for a large range of governing parameters and physical complexity.

Here, we have only examined relatively elementary aspects of methods for hyperbolic systems, but this short introduction should have taught you methods to solve such systems and introduced you to literature.

Increasingly we see methods developed for the inviscid Euler equation with shocks being used for the advection part of the Navier-Stokes solvers.