Weighted Essentially Non-Oscillatory Schemes

Author: Qiangqiang Shi
Advisor: Prof. Xiaolin Li

Department of Applied Mathematics and Statistics
Stony Brook University

September 25, 2012
Introduction

Basic formulation and WENO reconstruction

- Basic formulation
- The WENO reconstruction of $u_i^{-\frac{1}{2}}$, $u_i^{+\frac{1}{2}}$ from $\{\bar{u}_i\}$
- Monotone numerical flux
- Time discretization

Examples
Finite difference and related finite volume schemes are based on interpolations of discrete data using polynomials or other simple functions.

In the approximation theory, it is well known that the wider the stencil, the higher the order of accuracy of the interpolation, provided the function being interpolated is smooth inside the stencil.

Traditional finite difference methods are based on fixed stencil interpolations which works well for globally smooth problems.

However, fixed stencil interpolation of second or higher order accuracy is necessarily oscillatory near a discontinuity.

Such oscillations, which are called the Gibbs phenomena, do not decay in magnitude when the mesh is refined, and often leads to numerical instabilities in nonlinear problems containing discontinuities.
Before 1987, there were mainly two common ways to eliminate or reduce such oscillations near discontinuities.

- **Add an artificial viscosity,**
  - This could be tuned so that
    - Large enough near the discontinuity to suppress, or at least reduce the oscillations,
    - Small elsewhere to maintain high-order accuracy.

- **Disadvantage**
  - Fine tuning of the parameter controlling the size of the artificial viscosity is problem dependent.

- **Apply limiters to eliminate the oscillations.**
  - Reducing the order of accuracy of the interpolation near the discontinuity
    - Reducing the slope of linear interpolation,
    - Using a linear rather than a quadratic interpolation near the shock.

- **Disadvantage**
  - Accuracy necessarily degenerates to first order near smooth extrema.
WENO schemes are designed based on the successful ENO schemes by Harten et. al. in 1987.

The first WENO scheme is constructed by Liu, Chan and Osher in 1994 for a third order finite volume version in one space dimension.

In 1995, third and fifth order finite difference WENO scheme in multi space dimensions are constructed by Jiang and Shu, with a general framework for the design of the smoothness indicators and nonlinear weights.

Both ENO and WENO schemes use the idea of adaptive stencils in the reconstruction procedure based on the local smoothness of the numerical solution to automatically achieve high order and non-oscillatory property near discontinuities.
We consider the initial problems of nonlinear hyperbolic conservation laws:

\[
\begin{cases} 
    u_t + \nabla \cdot f(u) = 0 \\
    u(x,0) = u_0(x)
\end{cases}
\] (1)

Let \( \{l_i\} \) be a partition of \( R \), where \( l_i = [x_{i-\frac{1}{2}}, x_{j+\frac{1}{2}}] \) is the \( i \)-th cell. Denote \( \bar{u}_i(t) \) to be the sliding averages of the weak solution \( u(x,t) \) of (1) i.e.

\[
\bar{u}_i(t) = \frac{1}{\Delta x_i} \int_{l_i} u(x,t) \, dx
\] (2)

Integrating (1) over each cell \( l_i \), we obtain

\[
\frac{d}{dt} \bar{u}_i(t) + \frac{1}{\Delta x_i} (f(u(x_{i+\frac{1}{2}}, t)) - f(u(x_{i-\frac{1}{2}}, t))) = 0
\] (3)
Basic formulation (Cont.)

Replace the flux \( f(u(x_{i+\frac{1}{2}}, t)) \) with a monotone numerical flux \( \hat{f}(u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}) \), and get semi-discretization scheme:

\[
\frac{d}{dt} \bar{u}_i(t) + \frac{1}{\Delta x_i} (\hat{f}(u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}) - \hat{f}(u_{i-\frac{1}{2}}, u_{i-\frac{1}{2}})) = 0
\]  
(4)

where \( u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}} \) were reconstructed by WENO method, we can use Runge-Kutta method to solve the ODE (4).

Questions

- How to reconstruct \( u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}} \)?
- How to choose the monotone numerical flux \( \hat{f}(u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}) \)?
Reconstruct polynomials

- Reconstruct $k$-th degree polynomial $p_j(x)$, associated with each of the stencils $S_j$, $j = 0, \ldots, k$,
- Reconstruct $(2k)$-th degree polynomial $Q(x)$, associated with the larger stencil $T$.

such that:

$$
\bar{u}_{i+l} = \frac{1}{\Delta x_{i+l}} \int_{l_{i+l}} p_j(x) dx, \ l = -k + j, \ldots, j
$$

$$
\bar{u}_{i+l} = \frac{1}{\Delta x_{i+l}} \int_{l_{i+l}} Q(x) dx, \ l = -k, \ldots, k
$$
Compute the linear weights

Let

\[ Q(x_{i+\frac{1}{2}}) = \sum_{j=0}^{k} \gamma_j p_j(x_{i+\frac{1}{2}}) \]

For \( k = 2 \), the fifth order reconstruction, we have:

\[ p_0(x_{i+\frac{1}{2}}) = \frac{1}{3} \bar{u}_{i-2} - \frac{7}{6} \bar{u}_{i-1} + \frac{7}{6} \bar{u}_i \]

\[ p_1(x_{i+\frac{1}{2}}) = -\frac{1}{6} \bar{u}_{i-1} - \frac{5}{6} \bar{u}_i + \frac{1}{3} \bar{u}_{i+1} \]

\[ p_2(x_{i+\frac{1}{2}}) = \frac{1}{3} \bar{u}_i + \frac{5}{6} \bar{u}_{i+1} - \frac{1}{6} \bar{u}_{i+2} \]

\[ Q(x_{i+\frac{1}{2}}) = \frac{1}{30} \bar{u}_{i-2} - \frac{13}{60} \bar{u}_{i-1} + \frac{47}{60} \bar{u}_i + \frac{9}{20} \bar{u}_{i+1} - \frac{1}{20} \bar{u}_{i+2} \]

and we obtain the linear weights:

\[ \gamma_0 = \frac{1}{10}, \quad \gamma_1 = \frac{6}{10}, \quad \gamma_2 = \frac{3}{10} \]
Smoothness indicator, denoted by $\beta_j$, for each stencil $S_j$, which measures how smooth the function $p_j(x)$ is in the target cell $I_i$. The smaller this smoothness indicator $\beta_j$, the smoother the function $p_j(x)$ is in the target cell.

\[
\beta_j = \sum_{l=1}^{k} \int_{I_i} \Delta x_i^{2l-1} \left( \frac{\partial^l}{\partial x^l} p_j(x) \right)^2 dx
\] (5)

In the actual numerical implementation the smoothness indicators $\beta_j$ are written out explicitly as quadratic forms of the cell averages of $u$ in the stencil, for example when $k = 2$, we obtain:

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

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

\[
\beta_2 = \frac{13}{12} \left( \bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2} \right)^2 + \frac{1}{4} \left( \bar{u}_i - 4\bar{u}_{i+1} + \bar{u}_{i+2} \right)^2
\]
Based on the smoothness indicators we obtain the nonlinear weights $\omega_j$:

$$\omega_j = \frac{\bar{\omega}_j}{\sum_j \bar{\omega}_j}, \quad \bar{\omega}_j = \frac{\gamma_j}{\sum_j (\varepsilon + \beta_j)^2}$$

(6)

Here $\varepsilon$ is a positive real number (for example: $\varepsilon = 10^{-6}$) which is introduced to avoid the denominator to become zero.

The final WENO approximation is then given by:

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

(7)

- The reconstruction to $u_{i-\frac{1}{2}}^+$ is mirror symmetric with respect to $x_i$ of the above procedure.
- For systems of conservation laws, such as the Euler equation of gas dynamics, the reconstructions from $\{\bar{u}_i\}$ to $\{u_{i+\frac{1}{2}}^\pm\}$ are performed in the local characteristic directions to avoid oscillation.
The numerical flux $\hat{f}_{i+\frac{1}{2}}$ is defined by

$$\hat{f}_{i+\frac{1}{2}} = h(u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}^+)$$  \hspace{1cm} (8)$$

with the values $u_{i+\frac{1}{2}}^\pm$ obtained by the WENO reconstruction procedure.

The two argument function $h$ in (8) is a monotone flux. It satisfies:

- $h(a, b)$ is a Lipschitz continuous function in both arguments,
- $h(a, b)$ is a nondecreasing function in $a$ and a nonincreasing function in $b$. Symbolically $h(\uparrow, \downarrow)$,
- $h(a, b)$ is consistent with the physical flux $f$, that is, $h(a, a) = f(a)$. 

Author: Qiangqiang Shi  Advisor: Prof. Xiaolin Li
Monotone flux (Cont.)

Examples of monotone fluxes

- **Godunov flux:**
  \[
  h(a, b) = \begin{cases} 
  \min_{a \leq u \leq b} f(u) & \text{if } a < b \\
  \max_{b \leq u \leq a} f(u) & \text{if } a > b 
  \end{cases}
  \]

- **Engquist-Osher flux:**
  \[
  h(a, b) = \int_{0}^{a} \max(f'(u), 0) du + \int_{0}^{b} \min(f'(u), 0) du + f(0)
  \]

- **Lax-Friedrich flux:**
  \[
  h(a, b) = \frac{1}{2} [f(a) + f(b) - \alpha(b - a)]
  \]

  where \( \alpha = \max_u |f'(u)| \) is a constant. The maximum is taken over the relevant range of \( u \).
We have listed the monotone fluxes (Godunov flux, Engquist-Osher flux and Lax-Friedrich flux) from the least dissipative (less smearing of discontinuities) to the most.

For lower order methods (order of reconstruction is 1 or 2), there is a big difference between results obtained by different monotone fluxes. However, this difference becomes much smaller for higher order reconstructions.

We thus use the simple inexpensive Lax-Friedrich flux in most of the higher order calculations.
We write equation (4)

\[ \frac{d}{dt} \bar{u}_i(t) + \frac{1}{\Delta x_i} (\hat{f}(u_{i+\frac{1}{2}}, u_{i+\frac{1}{2}}) - \hat{f}(u_{i-\frac{1}{2}}, u_{i-\frac{1}{2}})) = 0 \]

as equation (9)

\[ u_t = L(u) \quad (9) \]

The third order TVD Runge-Kutta is simply:

\[ u^{(1)} = u^n + \Delta t L(u^n) \]

\[ u^{(2)} = \frac{3}{4} u^n + \frac{1}{4} u^{(1)} + \frac{1}{4} \Delta t L(u^{(1)}) \]

\[ u^{(3)} = \frac{1}{3} u^n + \frac{2}{3} u^{(2)} + \frac{2}{3} \Delta t L(u^{(1)}) \quad (10) \]
Another useful, although not TVD, fourth order Runge-Kutta scheme is:

\[
\begin{align*}
    u^{(1)} &= u^n + \frac{1}{2} \Delta tL(u^n) \\
    u^{(2)} &= u^n + \frac{1}{2} \Delta tL(u^{(1)}) \\
    u^{(3)} &= u^n + \Delta tL(u^{(2)}) \\
    u^{n+1} &= \frac{1}{3} (-u^n + u^{(1)} + 2u^{(2)} + u^{(3)}) + \frac{1}{6} \Delta tL(u^{(3)})
\end{align*}
\]  

(11)
We consider two well known problems which have the following Riemann type initial conditions:

\[ u(x, 0) = \begin{cases} 
  u_L & \text{if } x < 0 \\
  u_R & \text{if } x > 0 
\end{cases} \]

The first one is the Sod’s problem. The initial data are:

\[ (\rho_L, q_L, P_L) = (1, 0, 1) \quad ; \quad (\rho_R, q_R, P_R) = (0.125, 0, 0.1) \]

The Second one is the Riemann problem proposed by Lax:

\[ (\rho_L, q_L, P_L) = (0.445, 0.698, 3.528) \quad ; \quad (\rho_R, q_R, P_R) = (0.5, 0, 0.571) \]
Results

Lax problem results

Domain \((-1, 1)\), mesh size 160
Results

Lax problem results

Domain \((-1, 1)\), mesh size 320
Results

Sod problem results
Domain $(-1, 1)$, mesh size 160
Results

Sod problem results

Domain \((-1, 1)\), mesh size 320


Thanks for coming!