Lecture 2: Finite Difference WENO Schemes

Chi-Wang Shu

Division of Applied Mathematics
Brown University
Finite Difference WENO Schemes

Outline of the Second Lecture

• General description of conservative finite difference schemes for conservation laws

• Comparison with finite volume schemes

• An alternative flux formulation

• Inverse Lax-Wendroff procedure for boundary conditions

• Concluding remarks
Finite Difference WENO Schemes

Conservative finite difference schemes for conservation laws

A finite difference scheme approximates the conservation law

\[ u_t + f(u)_x = 0 \]

directly. The computational variables are the point values \( u_i \) of the solution, and the scheme is required to be in conservation form

\[ \frac{d}{dt} u_i + \frac{1}{\Delta x} \left( \hat{f}_{i+1/2} - \hat{f}_{i-1/2} \right) = 0 \]

where the numerical flux \( \hat{f}_{i+1/2} = \hat{f}(u_{i-p}, \cdots, u_{i+q}) \) is consistent with the physical flux \( \hat{f}(u, \cdots, u) = f(u) \) and is Lipschitz continuous with respect to all its arguments. We have assumed a uniform mesh \( \Delta x \) here for simplicity.
We would certainly desire

\[
\frac{1}{\Delta x} \left( \hat{f}_{i+1/2} - \hat{f}_{i-1/2} \right) = f(u) x \big|_{x=x_i} + O(\Delta x^r)
\]

for \( r \)-th order accuracy.

The following lemma (Shu and Osher, JCP88) establishes the relationship between the finite volume and finite difference schemes.

**Lemma:** If \( h(x) = h_{\Delta x}(x) \) is implicitly defined as

\[
\frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi) d\xi = f(u(x))
\]  
(1)

then

\[
\frac{1}{\Delta x} \left( h(x_{i+1/2}) - h(x_{i-1/2}) \right) = f(u) x \big|_{x=x_i}.
\]
The proof is straightforward: just take a $x$ derivative on both sides of (1).

This simple lemma indicates that we can take the numerical flux in the finite difference scheme as

$$\hat{f}_{i+1/2} = h(x_{i+1/2})$$

(2)

to ensure $r$-th order accuracy, if the function $h(x)$ in the Lemma can be computed to $r$-th order accuracy.
In fact, the (implicit) definition (1) of \( h(x) \) implies that

\[
\bar{h}_i \equiv \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} h(\xi) d\xi = f(u_i)
\]

is known for a finite difference scheme, since the point values \( u_i \) are the computational variables. Therefore, we are given the cell averages \( \bar{h}_i \) of the function \( h(x) \) and we would need to approximate its point values \( h(x_{i+1/2}) \) to high order accuracy to obtain the numerical flux \( \hat{f}_{i+1/2} \) in (2). Hence we can use the same reconstruction procedure discussed before, e.g. the WENO reconstruction, for finite volume schemes!
Finite Difference WENO Schemes

Same subroutine, different input

$\bar{u}_i$ for finite volume, $f(u_i)$ for finite difference

and different output

$u_{i+1/2}^\pm$ for finite volume, $\hat{f}_{i+1/2}$ for finite difference
For the purpose of stability, the finite difference procedure described above is applied to $f^+(u)$ and $f^-(u)$ separately, where $f^\pm(u)$ correspond to a flux splitting

$$f(u) = f^+(u) + f^-(u)$$

$$\frac{d}{du} f^+(u) \geq 0, \quad \frac{d}{du} f^-(u) \leq 0$$

The reconstruction for $f^+(u)$ uses a biased stencil with one more point to the left, and that for $f^-(u)$ uses a biased stencil with one more point to the right, to obey correct upwinding.
A commonly used flux splitting is the Lax-Friedrichs splitting

\[ f^\pm(u) = \frac{1}{2} \left( f(u) \pm \alpha u \right) \]

with

\[ \alpha = \max_u |f'(u)|. \]
Comparison of finite volume and finite difference schemes

- Finite volume schemes
  - are based on the cell averages $\{\bar{u}_i\}$, using an integral form
  - need a reconstruction $\{\bar{u}_i\} \rightarrow \{u_{i+1/2}^\pm\}$ (WENO)
  - can use any monotone flux $\hat{f}(u^-, u^+)$
  - do not need uniform or smooth meshes
  - larger computational cost for multi-dimensional problems
Finite Difference WENO Schemes

- Finite difference schemes
  - are based on the point values \( \{u_i\} \), using the PDE form
  - need a reconstruction \( \{f^\pm(u_i)\} \rightarrow \{\hat{f}^\pm_{i+1/2}\} \) (WENO)
  - can only use monotone fluxes which correspond to smooth flux splitting \( f(u) = f^+(u) + f^-(u) \) (however, see below)
  - need uniform or smooth meshes
  - smaller computational cost for multi-dimensional problems
Two (and multiple) spatial dimensions

The two dimensional conservation law:

\[ u_t + f(u)_x + g(u)_y = 0 \]
Finite volume method

Integrate over $I_{ij} \equiv [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$, we obtain

$$\frac{d \tilde{u}_{ij}(t)}{dt} = -\frac{1}{\Delta x_i \Delta y_j} \left( \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i+1/2}, y, t)) dy \right. \right.$$

$$- \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i-1/2}, y, t)) dy \left. \right)$$

$$+ \int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j+1/2}, t)) dx$$

$$- \int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j+1/2}, t)) dx$$
where $\tilde{u}$ is the cell average

$$\tilde{u}_{ij}(t) \equiv \frac{1}{\Delta x_i \Delta y_j} \int_{y_{i-1/2}}^{y_{i+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, y, t) \, dx \, dy$$

In particular, our notation is that $\bar{v}$ stands for the cell average in $x$:

$$\bar{v}_{ij} = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} v(x, y_j) \, dx$$

and $\tilde{v}$ stands for the cell average in $y$:

$$\tilde{v}_{ij} = \frac{1}{\Delta y_j} \int_{y_{j-1/2}}^{y_{j+1/2}} v(x_i, y) \, dy$$
This is approximated by the following finite volume scheme

\[ \frac{d\tilde{u}_{ij}(t)}{dt} = -\frac{1}{\Delta x_i}(\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j}) - \frac{1}{\Delta y_j}(\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2}) \]

where the numerical fluxes

\[ \hat{f}_{i+1/2,j} \approx -\frac{1}{\Delta y_j} \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i+1/2}, y, t)) dy \equiv \tilde{f}_{i+1/2,j} \]

\[ \hat{g}_{i,j+1/2} \approx -\frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j+1/2}, t)) dx \equiv \tilde{g}_{i,j+1/2} \]
First, look at the simple, linear constant coefficient case

\[ u_t + au_x + bu_y = 0 \]

we would have

\[ \hat{f}_{i+1/2,j} = a\tilde{u}_{i+1/2,j}, \quad \hat{g}_{i,j+1/2} = b\bar{u}_{i,j+1/2} \]

Therefore we would only need to do two one-dimensional reconstructions

\[ \{ \tilde{u}_{i,j} \} \rightarrow \{ \tilde{u}_{i+1/2,j} \} \quad \text{for fixed } j \]

\[ \{ \tilde{u}_{i,j} \} \rightarrow \{ \bar{u}_{i,j+1/2} \} \quad \text{for fixed } i \]

and the cost is the same as in the one-dimensional case per cell per direction.
However, if the PDE is nonlinear, namely if \( f(u) \) and \( g(u) \) are nonlinear functions of \( u \), then \( f(\tilde{u}) \neq \tilde{f}(u) \), hence we would need to do two one-dimensional reconstructions and one numerical integration (which bears about the same cost as a reconstruction) to obtain the numerical flux \( \hat{f}_{i+1/2,j} \). Thus we would need to do

\[
\{\tilde{u}_{ij}\} \rightarrow \{\tilde{u}_{i+1/2,j}\} \rightarrow \{u_{i+1/2,j+j_\alpha}\}_{\alpha=1}^{\alpha_g} \rightarrow \{\hat{f}_{i+1/2,j}\}
\]

where \( \{j + j_\alpha\}_{\alpha=1}^{\alpha_g} \) are the Gaussian quadrature points for the interval \([y_{j-1/2}, y_{j+1/2}]\) with sufficient high order of accuracy.

Likewise for \( \hat{g}_{i,j+1/2} \). This is now about three times the cost of the one-dimensional case per cell per direction.

This situation will be worse for three dimensions.
Finite difference method

The finite difference scheme for the two dimensional equation

\[ u_t + f(u)_x + g(u)_y = 0 \]

can proceed dimension by dimension. The scheme is

\[
\frac{du_{ij}(t)}{dt} = -\frac{1}{\Delta x}(\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j}) - \frac{1}{\Delta y}(\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2})
\]

where the numerical flux \( \hat{f}_{i+1/2,j} \) can be computed from \( \{u_{ij}\} \) with fixed \( j \) in exactly the same way as in the one dimensional case. Likewise for \( \hat{g}_{i,j+1/2} \). Therefore, the computational cost is exactly the same as in the one-dimensional case per point per direction.
Conclusions of comparison in multi-D

• In 2D, a finite volume scheme (of order of accuracy higher than 2) is 2 to 3 times as expensive as a finite difference scheme (of the same order of accuracy) using the same mesh and the same reconstruction procedure, depending on specific coding and type of computers.

• This discrepancy in cost is bigger for three dimensions.

• Finite volume schemes allow non-smooth and even unstructured meshes, while finite difference schemes would require uniform Cartesian or smooth curvilinear meshes.
An alternative flux formulation

The Shu-Osher lemma and the auxiliary function $h(x)$ defined above greatly simplify the computation of numerical fluxes for finite difference schemes, using the same reconstruction procedure as in finite volume schemes. However, this way of evaluating finite difference fluxes does have the following disadvantages:
Finite Difference WENO Schemes

- It can only be applied to a smooth flux splitting

\[ f(u) = f^+(u) + f^-(u) \]

where \( \frac{d}{du} f^+(u) \geq 0 \) (or in system case, this Jacobian matrix has only real and positive eigenvalues) and \( \frac{d}{du} f^-(u) \leq 0 \) (or in system case, this Jacobian matrix has only real and negative eigenvalues). In order to guarantee high order accuracy, the two functions \( f^+(u) \) and \( f^-(u) \) should be as smooth functions of \( u \) as \( f(u) \). The most common flux splitting used in finite difference schemes is therefore the Lax-Friedrichs flux splitting, which is the most diffusive among two-point monotone fluxes.
The Lax-Wendroff time discretization procedure (Qiu and Shu, SISC 2003) can be implemented only through repeated interpolation or reconstruction, resulting in a rather wide effective stencil. This is because we are directly performing reconstructions on $f(u)$ and hence do not have access to quantities like $u_x$.

It is difficult to maintain free stream solutions exactly in curvilinear meshes for multi-dimensional flow computation. This is because the fluxes in curvilinear coordinates involve metric derivatives, resulting in non-exact cancellations when nonlinear reconstructions are performed on $f(u)$ rather than on $u$. 
Recently, in Jiang, Shu and Zhang, SISC to appear, we explored an alternative flux formulation for finite difference WENO schemes which can overcome all three disadvantages above, at the price of a less clean and more costly procedure.

This alternative flux formulation was originally developed in Shu and Osher, JCP 1988. It was largely abandoned after the cleaner and less expensive formulation reviewed before appeared one year later in Shu and Osher, JCP 1989. We summarize the main ingredients of this alternative flux formulation below.
A major building block of the alternative flux formulation is a WENO interpolation (instead of reconstruction) procedure. Given the point values $u_i = u(x_i)$ of a piecewise smooth function $u(x)$, we would like to find an approximation of $u(x)$ at the half nodes $x_{i+\frac{1}{2}}$. The WENO interpolation procedure is very similar to the WENO reconstruction procedure, and has been developed and applied in, e.g., Sebastian and Shu, JSC 2003; Carlini, Ferretti and Russo, SISC 2005; Shu, SIAM Rev 2009.
It has been shown in Shu and Osher, JCP 1988 that there exist constants $a_2, a_4, \ldots$ such that

$$\hat{f}_{i+\frac{1}{2}} = f_{i+\frac{1}{2}} + \sum_{k=1}^{[(r-1)/2]} a_{2k} \Delta x^{2k} \left( \frac{\partial^{2k}}{\partial x^{2k}} f \right)_{i+\frac{1}{2}} + O(\Delta x^{r+1}) \quad (3)$$

which guarantees $k = r$-th order accuracy

$$\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) = f(u(x)) x \bigg|_{x_i} + O(\Delta x^k). \quad (4)$$

For fifth order accuracy,

$$\hat{f}_{i+\frac{1}{2}} = f_{i+\frac{1}{2}} - \frac{1}{24} \Delta x^2 f_{xx} \bigg|_{i+\frac{1}{2}} + \frac{7}{5760} \Delta x^4 f_{xxxx} \bigg|_{i+\frac{1}{2}}. \quad (5)$$
Finite Difference WENO Schemes

- The first term of the numerical flux in (5) is approximated by

\[ f_{i+\frac{1}{2}} = h(u_{i+\frac{1}{2}}^-, u_{i+\frac{1}{2}}^+) \]  

(6)

with the values \( u_{i+\frac{1}{2}}^\pm \) obtained by the WENO interpolation. Here \( h \) can be any monotone flux or approximate Riemann solver, without the flux splitting and smoothness assumption as before.

- The remaining terms of the numerical flux in (5) have at least \( \Delta x^2 \) in their coefficients, hence they only need lower order approximations and they are expected to contribute much less to spurious oscillations. Therefore, following the practice in Qiu and Shu, SISC 2003, we approximate these remaining terms by simple central approximation or one-point upwind-biased approximation with suitable orders of accuracy, without using the more expensive WENO procedure.
For finite difference schemes approximating PDEs, there are two major difficulties associated with numerical boundary conditions:

- High order finite difference schemes involve a wide stencil, hence there are several points near the boundary (either as ghost points outside the computational domain or as the first few points inside the computational domain near the boundary) which need different treatment.
The boundary of the computational domain may not coincide with grid points.

For example, in 1D, we may have the physical boundary \( x = 0 \) located anywhere between two grid points. While this seems artificial, it is unavoidable for a moving boundary computed on a fixed grid.

This difficulty is more profound in 2D (complicated geometry computed on Cartesian meshes).

One of the major difficulties is the small cell near the boundary and the resulting small time step required for stability.
Previous work on numerical boundary conditions:

- $h$-box method of Berger, Helzel and LeVeque (SINUM 2003): suitable flux computation based on cells of size $h$. This method can overcome the difficulty of small time step for stability, but is somewhat complicated in 2D and for high order accuracy.

- Reflecting or symmetry boundary conditions for ghost points: suitable for solid walls or symmetry lines which are straight lines but lead to large errors for curved walls not aligned with meshes.
Finite Difference WENO Schemes

- Extrapolation to obtain ghost point values (Kreiss et al. SINUM 2002, 2004; SISC 2006; Sjögreen and Petersson CiCP 2007). A GKS stability analysis must be performed to assess its stability. Second order is fine but higher order is more complicated to analyze. It is not stable if the physical boundary is too close to a grid point.

- Converting spatial derivative near the boundary to temporal derivatives (Goldberg and Tadmor, Math Comp 1978, 1981 for one-dimensional linear hyperbolic initial-boundary value problems).
Review on the traditional Lax-Wendroff procedure for solving, e.g.

\[ u_t + u_x = 0 \]

- Taylor expansion in time

\[ u_j^{n+1} = u_j + (u_t)_j \Delta t + \frac{1}{2}(u_{tt})_j \Delta t^2 + \ldots \]

- Replace the time derivatives by spatial derivatives by repeatedly using the PDE:

\[ (u_t)_j = -(u_x)_j \]
\[ (u_{tt})_j = -((u_x)_t)_j = -((u_t)_x)_j = (u_{xx})_j \]

\[ \ldots \]

- Approximate the spatial derivatives by finite differences of suitable order of accuracy.
We now look at the basic idea of the inverse Lax-Wendroff procedure, by switching the roles of $x$ and $t$ in the traditional Lax-Wendroff procedure. Suppose we are solving

$$u_t + u_x = 0, \quad u(0, t) = g(t)$$

and suppose the boundary $x = 0$ is of distance $a\Delta x$ from $x_1$ (with a constant $a$), the inverse Lax-Wendroff procedure to determine $u_1$ is as follows:
Finite Difference WENO Schemes

- Taylor expansion in space

\[ u_1 = u(0, t) + u_x(0, t)a \Delta x + \frac{1}{2} u_{xx}(0, t)(a \Delta x)^2 + ... \]

- Replace the spatial derivatives by time derivatives by repeatedly using the PDE:

\[ u_x = -u_t; \quad u_x(0, t) = -u_t(0, t) = -g'(t) \]

\[ u_{xx} = (-u_t)_x = -(u_x)_t = u_{tt}; \]

\[ u_{xx}(0, t) = u_{tt}(0, t) = g''(t) \]

... 

- Compute \( g'(t) \), \( g''(t) \), etc. either analytically or by finite difference.
To best illustrate the idea of the inverse Lax-Wendroff type procedure, we use 1D scalar conservation laws as an example

\[
\begin{cases}
    u_t + f(u)_x = 0 & x \in (-1, 1), \quad t > 0, \\
    u(-1, t) = g(t) & t > 0, \\
    u(x, 0) = u_0(x) & x \in [-1, 1].
\end{cases}
\]  

(7)

We assume \( f'(u(-1, t)) \geq \alpha > 0 \) and \( f'(u(1, t)) \geq \alpha > 0 \) for \( t > 0 \). This assumption guarantees the left boundary \( x = -1 \) is an inflow boundary where a boundary condition is needed and the right boundary \( x = 1 \) is an outflow boundary where no boundary condition is needed.
Let us discretize the interval $(-1, 1)$ by a uniform mesh

$$-1 + \Delta x/2 = x_0 < x_1 < \cdots < x_N = 1 - \Delta x/2. \quad (8)$$

Notice that both $x_0$ and $x_N$ are not located on the boundary.

At the inflow boundary $x = -1$, a Taylor expansion of order $s - 1$ gives

$$u(x_j, t_n) = \sum_{k=0}^{s-1} \frac{(x_j + 1)^k}{k!} \left. \frac{\partial^k u}{\partial x^k} \right|_{x=-1, t=t_n} + O(\Delta x^s),$$

for $j = -1, -2, -3$. Hence a $s$-th order approximation of the values $u_j$ at the ghost points is

$$u_j = \sum_{k=0}^{s-1} \frac{(x_j + 1)^k}{k!} \left. \frac{\partial^k u}{\partial x^k} \right|_{x=-1, t=t_n}, \quad j = -1, -2, -3. \quad (9)$$
Here we suppress the $t_n$ dependence on the left hand side. We already have $u(-1, t_n) = g(t_n)$. To obtain the spatial derivatives, we utilize the PDE

$$u_t + f'(u)u_x = 0$$

and evaluate it at $x = -1, t = t_n$. We have

$$u_x(-1, t_n) = -\frac{u_t(-1, t_n)}{f'(u(-1, t_n))} = -\frac{g'(t_n)}{f'(g(t_n))},$$

where $f'(g(t_n))$ is bounded away from zero by the assumption that $x = -1$ is an inflow boundary.
Differentiating the PDE with respect to time yields

\[ u_{tt} + f''(u) u_t u_x + f'(u) u_{xt} = 0. \]  \hspace{1cm} (10)

The term \( u_{xt} \) can be written as

\[
\begin{align*}
    u_{xt} &= (u_t)_x \\
    &= -(f'(u)u_x)_x \\
    &= -f''(u)u_x^2 - f'(u)u_{xx}.
\end{align*}
\]

Substituting it into (10), we obtain an equation for \( u_{xx} \)

\[ u_{tt} + f''(u) u_t u_x - f'(u) f''(u) u_x^2 - f'(u)^2 u_{xx} = 0. \]  \hspace{1cm} (11)
Solving (11) for \( u_{xx} \) and evaluating it at \( x = -1, t = t_n \), we have

\[
\begin{align*}
  u_{xx}(-1, t_n) &= \frac{g''(t_n) + f''(g(t_n))g'(t_n)u_x(-1, t_n) - f'(g(t_n)) f''(g(t_n))}{f'(g(t_n))^2} \\
  &= \frac{f'(g(t_n)) g''(t_n) - 2f''(g(t_n)) g'(t_n)^2}{f'(g(t_n))^3}.
\end{align*}
\]

Following the same procedure, we can obtain values of \( \frac{\partial^k u}{\partial x^k} \bigg|_{x=-1,t=t_n} \), \( k = 1, \cdots, s - 1 \).

At the outflow boundary \( x = 1 \), extrapolation of appropriate order is used. Either a regular or a WENO type extrapolation is appropriate depending on whether the outflow solution is smooth or contains shocks.
We have proved the linear stability of our numerical boundary conditions for linear wave equations according to the theory of Gustafsson, Kreiss and Sundström (GKS), in the framework of semi-discrete schemes as studied in Strikwerda (JCP 1980).

For the outflow boundary condition, we can show that the semi-discrete problem with the extrapolation is stable for all order $s$.

We remark that the time step restriction of solving the system of ODEs with our boundary treatment is not more severe than the pure initial value problem according to our computational experience. The standard CFL conditions determined by the interior schemes are used in the numerical examples.
For characteristic or nearly characteristic boundary condition, a least square procedure combining the inverse Lax-Wendroff procedure and extrapolation can be used.

For systems, the procedure is performed on local characteristic fields, combining the inverse Lax-Wendroff procedure for inflow and extrapolation for outflow.

The same procedure has been developed in 2D for various boundary conditions including solid walls which may be curved. The computational domain does not need to be aligned with the mesh.

A simplified and improved version with the relatively complicated inverse Lax-Wendroff procedure applied only to the first order normal derivative maintains the same stability and accuracy properties in numerical tests.
**Example 1.** We test the Burgers equation

\[
\begin{aligned}
    u_t + \left( \frac{1}{2} u^2 \right)_x &= 0 & x &\in (-1, 1), & t &> 0, \\
    u(x, 0) &= 0.25 + 0.5 \sin(\pi x) & x &\in [-1, 1], \\
    u(-1, t) &= g(t) & t &> 0.
\end{aligned}
\]

Here \( g(t) = w(-1, t) \), where \( w(x, t) \) is the exact solution of the initial value problem on \((-1, 1)\) with periodic boundary conditions. For all \( t \), the left boundary \( x = -1 \) is an inflow boundary and the right boundary \( x = 1 \) is an outflow boundary.
Table 1: Errors of the Burgers equation (12). $\Delta x = 2/N$ and $t = 0.3$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>9.11E-05</td>
<td></td>
<td>3.56E-04</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>3.10E-06</td>
<td>4.88</td>
<td>1.35E-05</td>
<td>4.72</td>
</tr>
<tr>
<td>160</td>
<td>1.31E-07</td>
<td>4.57</td>
<td>6.51E-07</td>
<td>4.38</td>
</tr>
<tr>
<td>320</td>
<td>3.97E-09</td>
<td>5.05</td>
<td>2.68E-08</td>
<td>4.60</td>
</tr>
<tr>
<td>640</td>
<td>1.02E-10</td>
<td>5.29</td>
<td>8.34E-10</td>
<td>5.00</td>
</tr>
<tr>
<td>1280</td>
<td>2.86E-12</td>
<td>5.15</td>
<td>2.62E-11</td>
<td>5.00</td>
</tr>
</tbody>
</table>
Figure 1: Burgers equation (12), $\Delta x = 1/40$. Solid line: exact solution; Symbols: numerical solution.
Example 2. Euler equations, blast wave example. We consider the interaction of two blast waves. The initial data are

\[
U(x, 0) = \begin{cases} 
U_L & 0 < x < 0.1, \\
U_M & 0.1 < x < 0.9, \\
U_R & 0.9 < x < 1,
\end{cases}
\]

where \( \rho_L = \rho_M = \rho_R = 1, u_L = u_M = u_R = 0, \)
\( p_L = 10^3, p_M = 10^{-2}, p_R = 10^2. \) There are solid wall boundary conditions at both \( x = 0 \) and \( x = 1. \) This problem involves multiple reflections of shocks and rarefactions off the walls. There are also multiple interactions of shocks and rarefactions with each other and with contact discontinuities.
Figure 2: The density profiles of the blast wave problem. Solid lines: reference solution computed by the fifth order WENO scheme with $\Delta x = 1/16000$; Symbols: numerical solutions by our boundary treatment.
Example 3. We test the 2D Burgers equation

\[
\begin{aligned}
    u_t + \frac{1}{2} (u^2)_x + \frac{1}{2} (u^2)_y &= 0 & (x, y) \in \Omega, & t > 0, \\
    u(x, y, 0) &= 0.75 + 0.5 \sin [\pi (x + y)] & (x, y) \in \bar{\Omega}, \\
    u(x, y, t) &= g(x, y, t) & (x, y) \in \Gamma, & t > 0,
\end{aligned}
\]  

(13)

where

\[
\Omega = (-1, 1) \times (-1, 1), \\
\Gamma = \{(x, y) : x = -1 \text{ or } y = -1\},
\]

or

\[
\Omega = \{(x, y) : x^2 + y^2 < 0.5\}, \\
\Gamma = \{(x, y) : x^2 + y^2 = 0.5 \text{ and } x + y \leq 0\}.
\]
Here \( g(x, y, t) = w(x, y, t) \), where \( w(x, y, t) \) is the exact solution of the initial value problem on \((-1, 1) \times (-1, 1)\) with periodic boundary conditions. Notice that in the second case the domain boundary is not aligned with the Cartesian meshes.

Table 2: Errors of the 2D Burgers equation (13). \( \Delta x = 2/N_x, \Delta y = 2/N_y, t = 0.15 \).

| \( N_x = N_y \) | on a square | on a disk |
|-----------------|-------------|-----------------|-----------------|-----------------|
| \( L^1 \) error | order | \( L^\infty \) error | order | \( L^1 \) error | order | \( L^\infty \) error | order |
| 40              | 1.55E-04 | 9.86E-03 | 1.10E-04 | 1.77E-03 |
| 80              | 1.06E-05 | 3.87 | 1.80E-03 | 2.46 | 7.24E-06 | 3.93 | 4.06E-04 | 2.12 |
| 160             | 4.93E-07 | 4.43 | 2.38E-04 | 2.91 | 4.65E-07 | 3.96 | 4.77E-05 | 3.09 |
| 320             | 3.47E-08 | 3.83 | 2.83E-05 | 3.08 | 3.63E-08 | 3.68 | 6.04E-06 | 2.98 |
| 640             | 2.72E-09 | 3.67 | 2.85E-06 | 3.31 | 4.10E-09 | 3.15 | 9.45E-07 | 2.68 |
Figure 3: 2D Burgers equation (13). \( \Delta x = \Delta y = 1/40 \). Cut along the diagonal. Solid line: exact solution; Symbols: numerical solution.
(a) on a disk, $t = 0.55$

(b) on a disk, $t = 6$

Figure 4: Continued.
Example 4. We are most interested in applying our method to the solid wall boundary conditions \((u, v) \cdot n = 0\), when the wall is not aligned with the grid and can be curved. Our first example of this kind is the double Mach reflection problem. This problem is initialized by sending a horizontally moving shock into a wedge inclined by a \(30^\circ\) angle. In order to impose the solid wall condition by the reflection technique, people usually solve an equivalent problem that puts the solid wall horizontal and puts the shock \(60^\circ\) angle inclined to the wall. Another way to avoid the trouble of imposing boundary conditions is to use a multidomain WENO method. With the use of our method, we are able to solve the original problem with a uniform mesh in a single domain.
Figure 5: Left: The computational domain (solid line). The dashed line indicates the computational domain used in the traditional finite difference solvers. The square points indicate some of the grid points. Right: Density contour of double Mach reflection. $\Delta x = \Delta y = \frac{1}{320}$. 

Division of Applied Mathematics, Brown University
Figure 6: Density contours of double Mach reflection, 30 contours from 1.731 to 20.92. Zoomed-in near the double Mach stem. The plots in the left column (our computation with the new boundary condition treatment) are rotated and translated for comparison.
(a) $\Delta x = \Delta y = \frac{1}{640}$, original problem  
(b) $\Delta x = \Delta y = \frac{\sqrt{3}}{960}$, equivalent problem

Figure 7: Continued
Example 5. This example involves a curved wall which is a circular cylinder of unit radius positioned at the origin on a $x$-$y$ plane. The problem is initialized by a Mach 3 flow moving toward the cylinder from the left. In order to impose the solid wall boundary condition at the surface of the cylinder by the reflection technique, a particular mapping from the unit square to the physical domain is usually used in traditional finite difference methods. Using our method, we are able to solve this problem directly in the physical domain.
Figure 8: Physical domain of flow past a cylinder. The square points indicate some of the grid points near the cylinder. Illustrative sketch, not to scale.
Figure 9: Pressure contour of flow past a cylinder.

(a) \( \Delta x = \Delta y = \frac{1}{20} \)

(b) \( \Delta x = \Delta y = \frac{1}{40} \)
References:


We extend the high order accurate numerical boundary condition based on finite difference methods to simulations of compressible inviscid flows involving complex moving geometries.

- For problems in such geometries, it is difficult to use body-fitted meshes which conform to the moving geometry.

- Instead, methods based on fixed Cartesian meshes have been successfully developed. For example, the immersed boundary (IB) method introduced by Peskin (JCP 1972) is widely used. One of the challenges of the IB method is the representation of the moving objects which cut through the grid lines in an arbitrary fashion.
Finite Difference WENO Schemes

• To solve compressible inviscid flows in complex moving geometries, most methods in the literature are based on finite volume schemes. The challenge mainly comes from the so-called “small-cell” problem. Namely, one obtains irregular cut cells near the boundary, which may be orders of magnitude smaller than the regular grid cells, leading to a severe time step restriction.

• In terms of accuracy, most finite volume schemes in the literature are at most second order. In particular, the errors at the boundaries sometimes often fall short of second order.

• Our inverse Lax-Wendroff procedure can be extended to such situations with moving geometries. The only change is to obtain relationships between the temporal and spatial derivatives via the PDE in a moving Lagrangian framework.
Example 6. We consider a gas confined between two rigid walls. The right wall is fixed at $x_r = 1.0$ while the left wall is moving. We assume the left wall is positioned at $x_l(t) = 0.5(1 - t)$. The initial conditions are

\[
\begin{align*}
\rho(x, 0) &= 1 + 0.2 \cos \left[2\pi \left(x - 0.5\right)\right], \\
u(x, 0) &= x - 1, \\
p(x, 0) &= \rho(x, 0)^\gamma,
\end{align*}
\]

such that the initial entropy $s(x, 0) = 1$. As long as the solution stays smooth, we have isentropic flow, i.e., $s(x, t) = 1$. Thus the numerical value of the entropy can be used for the analysis of convergence.
Table 3: Entropy errors and convergence rates of Example 9

<table>
<thead>
<tr>
<th>$h$</th>
<th>$x_l(t) = 0.5(1 - \sin t)$</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/40</td>
<td>7.26E-07</td>
<td>1.32E-06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1/80</td>
<td>1.15E-08</td>
<td>5.98</td>
<td>5.55</td>
<td>2.82E-08</td>
<td>5.55</td>
</tr>
<tr>
<td>1/160</td>
<td>3.43E-10</td>
<td>5.07</td>
<td>5.51</td>
<td>6.19E-10</td>
<td>5.51</td>
</tr>
<tr>
<td>1/320</td>
<td>9.90E-12</td>
<td>5.11</td>
<td>4.64</td>
<td>2.49E-11</td>
<td></td>
</tr>
</tbody>
</table>
Example 7. This is a 1D problem involving shocks and rarefaction waves. A piston with width $10h$ is initially centered at $x = -5h$ inside a shock tube. Here $h$ is the mesh size. The piston instantaneously moves with a constant velocity $u_p = 2$ into an initially quiescent fluid with $\rho = 1$ and $p = 5/7$. This problem is equivalent to two independent Riemann problems and thus the exact solution can be obtained. A shock forms ahead of the piston and a rarefaction wave forms in the rear.
Figure 10: Density and pressure profiles of Example 10. The piston is represented by the rectangle. Solid lines: exact solutions; Symbols: numerical solutions with $h = 0.25$. 

Division of Applied Mathematics, Brown University
Example 8. Our next example involves 2D flows in complex moving geometries. The computational domain is $[-4, 4] \times [-4, 4]$ with all the boundaries as rigid walls. A rigid cylinder with radius $R = 1$ is initially centered at $(0, 0)$ and starts moving. The center of the cylinder is positioned at $X_c(t)$. We use our high order boundary treatment at the surface of the moving cylinder and the reflection technique at the fixed walls.

In our first case, we take $X_c = (-0.5 \sin t, 0)$ such that the cylinder moves horizontally. In the second case, we take $X_c = (-0.5 \sin t, 0.3t)$ such that the cylinder moves in the 2D space.
Figure 11: Density contours of Example 12. $h = 1/40$, $t = 0.4$.

(a) $X_c = (-0.5 \sin t, 0)$

(b) $X_c = (-0.5 \sin t, 0.3t)$
Table 4: Entropy errors and convergence rates of Example 12. $t = 0.4$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$x_c = (-0.5 \sin t, 0), t = 0.7$</th>
<th>$x_c = (-0.5 \sin t, 0.3t), t = 0.5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$L^1$ error</td>
<td>order</td>
</tr>
<tr>
<td>1/5</td>
<td>4.11E-03</td>
<td></td>
</tr>
<tr>
<td>1/10</td>
<td>3.86E-04</td>
<td>3.41</td>
</tr>
<tr>
<td>1/20</td>
<td>1.21E-05</td>
<td>5.00</td>
</tr>
<tr>
<td>1/40</td>
<td>2.43E-07</td>
<td>5.64</td>
</tr>
</tbody>
</table>
Example 9. The last example shows that our high order method can also treat a rigid body whose motion is induced by the fluid. We test the so-called cylinder lift-off problem. In this problem, a rigid cylinder initially resting on the floor of a 2D channel is driven and lifted by a strong shock. The computational domain is $[0, 1] \times [0, 0.2]$. A rigid cylinder with radius 0.05 and density 10.77 is initially centered at $(0.15, 0.05)$. A Mach 3 shock starts at $x = 0.08$ moving towards the cylinder. The density and pressure of the resting gas are $\rho = 1.4$ and $p = 1.0$ respectively. The top and bottom of the domain are rigid walls. The left boundary is set to the post-shock state and the right boundary is supersonic outflow.
Table 5: Center of the cylinder of Example 9.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$t = 0.1641$</th>
<th>$t = 0.30085$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$-coordinate</td>
<td>$y$-coordinate</td>
</tr>
<tr>
<td>1/160</td>
<td>3.7058E-01</td>
<td>8.1140E-02</td>
</tr>
<tr>
<td>1/320</td>
<td>3.6153E-01</td>
<td>8.3219E-02</td>
</tr>
<tr>
<td>1/640</td>
<td>3.5706E-01</td>
<td>8.3680E-02</td>
</tr>
<tr>
<td>1/1280</td>
<td>3.5539E-01</td>
<td>8.4133E-02</td>
</tr>
<tr>
<td>1/2560</td>
<td>3.5461E-01</td>
<td>8.4258E-02</td>
</tr>
</tbody>
</table>
Figure 12: Pressure contours at \( t = 0.1641 \). 53 contours from 2 to 28.

(a) \( h = 1/640 \)

(b) \( h = 1/1280 \)
Figure 13: Pressure contours at $t = 0.30085$. 53 contours from 2 to 28. $t = 0.30085$. 

Division of Applied Mathematics, Brown University
References:


Concluding remarks

- High order conservative finite difference schemes can only be obtained on structured smooth meshes, but they are very simple and efficient for multi-dimensional computations.

- WENO reconstruction or interpolation in finite difference schemes can provide high order accuracy and essentially non-oscillatory shock transition.
• An alternative flux formulation can provide a finite difference scheme suitable for any monotone fluxes or approximate Riemann solvers, has a narrow stencil with Lax-Wendroff time discretization, and can maintain free-stream solutions on curvilinear meshes, at the price of a less clean and more costly procedure.

• An inverse Lax-Wendroff procedure for boundary treatment can allow the the boundary of computational domain not aligned with the grids, and also for fixed structured grid for problems with moving objects, without sacrificing stability or accuracy.
The End

THANK YOU!