WENO finite volume and finite difference schemes

Chi-Wang Shu

Division of Applied Mathematics

Brown University

FlowModellium Lab, MIPT, September 2013

Outline

- Introduction
- The WENO interpolation and reconstruction procedure
- WENO finite volume schemes for conservation laws
- Time discretization
- WENO finite difference schemes for conservation laws
- Comparison of finite volume and finite difference schemes
- Two (and multiple) spatial dimensions
- Generalization to systems

- Recent development and applications

1. Inverse Lax-Wendroff type boundary treatments
2. Free-stream preserving finite difference schemes on curvilinear meshes
3. A homotopy method based on WENO schemes for solving steady state problems
4. Application: Shock-vortex and vortex-vortex interactions

Introduction

The purpose of the initial design of WENO schemes was to solve the hyperbolic conservation law

$$u_t + f(u)_x = 0$$

which has discontinuous solutions even if the initial condition is smooth.

We would like to have schemes which are both

- high order accurate in smooth regions, and
- (essentially) non-oscillatory with sharp shock transition

For typical linear schemes, i.e. schemes which are linear for a linear PDE

$$u_t + au_x = 0 \tag{1}$$

for example the second order accurate Lax-Wendroff scheme

$$u_j^{n+1} = \frac{a\lambda}{2}(1 + a\lambda)u_{j-1}^n + (1 - a^2\lambda^2)u_j^n - \frac{a\lambda}{2}(1 - a\lambda)u_{j+1}^n$$

where $\lambda = \frac{\Delta t}{\Delta x}$ and $|a|\lambda \leq 1$, the two properties desired above cannot be fulfilled simultaneously (Godunov Theorem).

We would therefore need to consider *nonlinear* schemes, which are nonlinear even for the linear PDE (1), such as the ENO and WENO schemes.

Essentially non-oscillatory (ENO) schemes:

- Finite volume ENO schemes: Harten, Engquist, Osher and Chakravarthy, JCP 1987. (SCI cited 1,126 times)
- Finite difference ENO schemes: Shu and Osher, JCP 1988, 1989. (SCI cited 1,350 and 925 times respectively)
- Uniform high order polynomial interpolation or reconstruction for the numerical fluxes (or using other basis functions)
- The stencil is locally adaptive: among several candidate stencils *one* is chosen according to local smoothness.

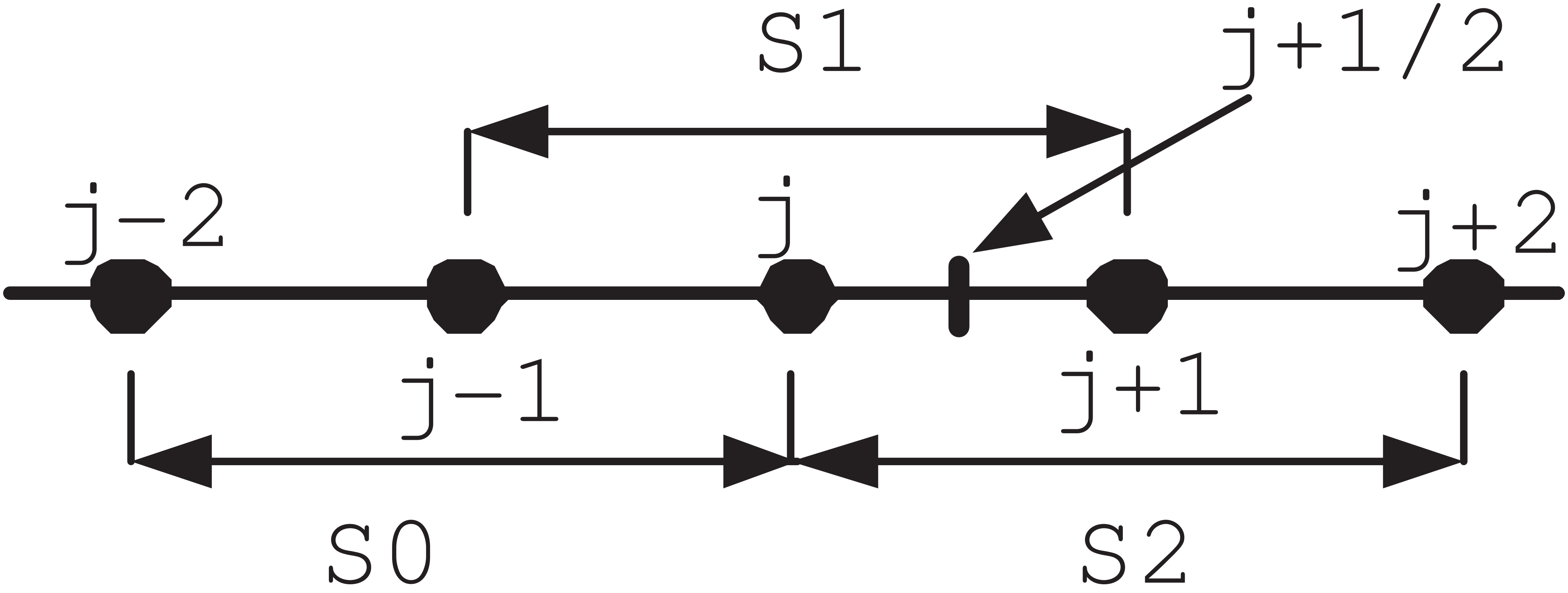


Figure 1: Three possible stencils for approximating the flux at $x_{j+1/2}$ using three points in each stencil.

Weighted ENO (WENO) schemes:

- Third order finite volume WENO scheme: Liu, Osher and Chan, JCP 1994 (SCI cited 653 times)
- Fifth order finite difference and general framework for the WENO schemes: Jiang and Shu, JCP 1996. (SCI cited 1,215 times)
- Instead of using just *one* candidate stencil; a linear combination of *all* candidate stencils is used
- The choice of the weight to each candidate stencil, which is a *nonlinear* function of the grid values, is a key to the success of WENO.

Advantages of ENO and WENO schemes:

- Uniform high order accuracy in smooth regions including at smooth extrema, unlike second order TVD schemes which degenerate to first order accuracy at smooth extrema
- Sharp and essentially non-oscillatory (to the eyes) shock transition
- Robust for many physical systems with strong shocks
- Especially suitable for simulating solutions containing both discontinuities and complicated smooth solution structure, such as shock interaction with vortices

Some advantages of WENO schemes over ENO schemes:

- Higher order of accuracy with the same set of candidate stencils: the order of accuracy is 3 instead of 2 for piecewise linear, and 5 instead of 3 for piecewise quadratic
- No logical “if” statements in the stencil choosing process of ENO. Cleaner programming
- Numerical flux function is smoother: C^∞ instead of only Lipschitz as in the ENO case. Hence: (i) a convergence proof when the solution is smooth, and (ii) better steady state convergence.

The WENO interpolation and reconstruction procedure

Problem 1: interpolation. Given the point values $u_i = u(x_i)$ of a piecewise smooth function $u(x)$ at the grid points x_i, find an approximation to the function $u(x)$ or its derivative $u'(x)$ at a desired location, e.g. at the half-nodes $x_{i+1/2}$.

Problem 2: reconstruction. Given the cell averages $\bar{u}_i = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x)dx$ of a piecewise smooth function $u(x)$ for the cells $I_i = [x_{i-1/2}, x_{i+1/2}]$ with cell sizes Δx_i, find an approximation to the function $u(x)$ at a desired location, e.g. at the cell boundaries $x_{i+1/2}$.

We will concentrate our discussion on **Problem 2** since it is more relevant to the solution of conservation laws. However, Problem 1 is also useful for situations such as solutions to Hamilton-Jacobi equations.

General procedure of reconstruction with a given stencil. e.g. to reconstruct $u_{i+1/2}$ given $\bar{u}_{i-1}$, $\bar{u}_i$ and $\bar{u}_{i+1}$:

1. Find the unique second order polynomial $p(x)$ which agrees with the three given cell averages $\bar{u}_{i-1}$, $\bar{u}_i$ and $\bar{u}_{i+1}$ for the three cells in the stencil, respectively:

$$\frac{1}{\Delta x_{i-1}} \int_{x_{i-3/2}}^{x_{i-1/2}} p(x)dx = \bar{u}_{i-1},$$

$$\frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} p(x)dx = \bar{u}_i,$$

$$\frac{1}{\Delta x_{i+1}} \int_{x_{i+1/2}}^{x_{i+3/2}} p(x)dx = \bar{u}_{i+1}.$$

2. Take the value $p(x_{i+1/2})$ as an approximation to $u_{i+1/2}$:

$$u_{i+1/2} = p(x_{i+1/2})$$

3. The approximation $u_{i+1/2}$ can be written out eventually as a linear combination of the given cell averages $\bar{u}_{i-1}$, $\bar{u}_i$ and $\bar{u}_{i+1}$ because the procedure is linear:

$$u_{i+1/2} = -\frac{1}{6}\bar{u}_{i-1} + \frac{5}{6}\bar{u}_i + \frac{1}{3}\bar{u}_{i+1}$$

This approximation is third order accurate if the function $u(x)$ is smooth in the stencil $\{I_{i-1},\ I_i,\ I_{i+1}\}$.

Using such approximations with a fixed stencil leads to high order linear schemes, which will be oscillatory in the presence of shocks by the Godunov Theorem.

The general procedure of a WENO reconstruction:

1. Compute the approximations from several different substencils, e.g. the three stencils in Figure 2:

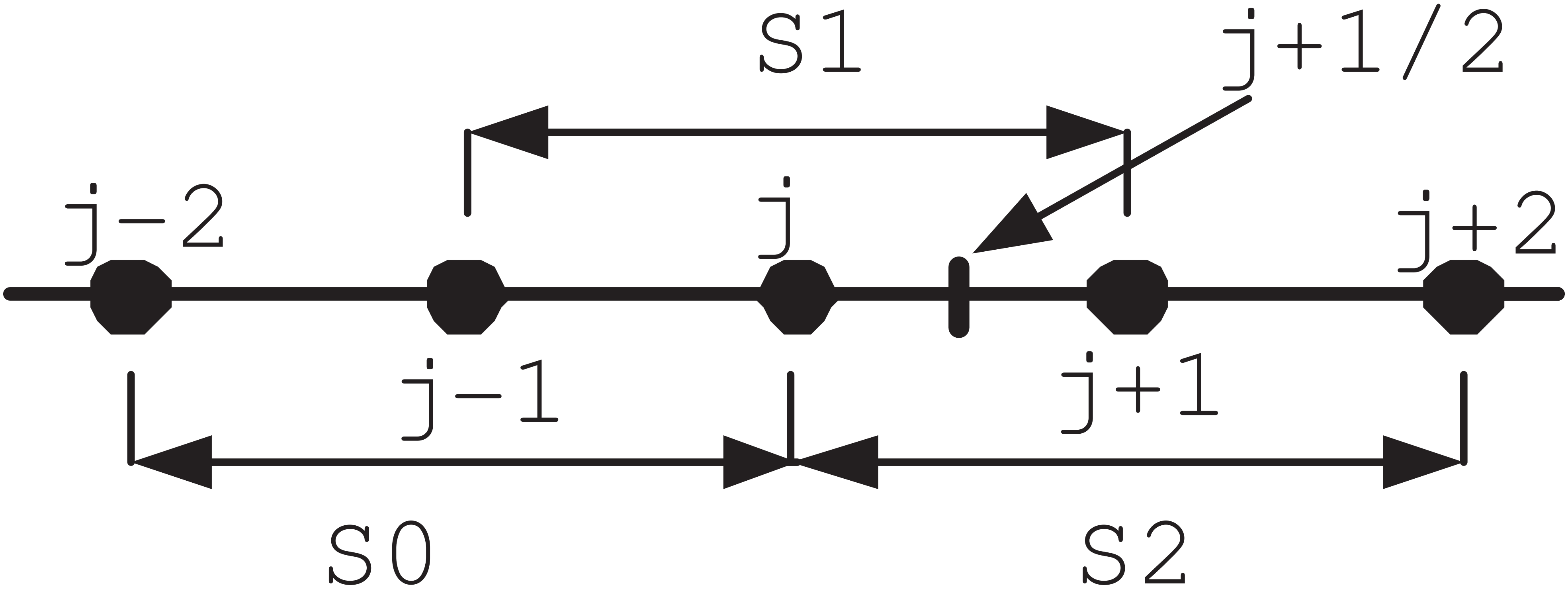


Figure 2: Three sub-stencils for the reconstruction at $x_{j+1/2}$ using three cells in each stencil.

$$u^{(0)}_{i+1/2} = \frac{1}{3}\bar{u}_{i-2} - \frac{7}{6}\bar{u}_{i-1} + \frac{11}{6}\bar{u}_{i}$$

$$u^{(1)}_{i+1/2} = -\frac{1}{6}\bar{u}_{i-1} + \frac{5}{6}\bar{u}_{i} + \frac{1}{3}\bar{u}_{i+1}$$

$$u^{(2)}_{i+1/2} = \frac{1}{3}\bar{u}_{i} + \frac{5}{6}\bar{u}_{i+1} - \frac{1}{6}\bar{u}_{i+2}$$

If the function $u(x)$ is smooth in all three substencils, then the three approximations $u^{(0)}_{i+1/2}$, $u^{(1)}_{i+1/2}$ and $u^{(2)}_{i+1/2}$ are all third order accurate.

2. Find the combination coefficients γ_0, γ_1 and γ_2, also called *linear weights*, such that the linear combination

$$u_{i+1/2} = \gamma_0 u_{i+1/2}^{(0)} + \gamma_1 u_{i+1/2}^{(1)} + \gamma_2 u_{i+1/2}^{(2)}$$

is fifth order accurate if $u(x)$ is smooth in all substencils. This can be easily achieved with $\gamma_0 = \frac{1}{10}$, $\gamma_1 = \frac{3}{5}$ and $\gamma_2 = \frac{3}{10}$:

$$u_{i+1/2} = \frac{1}{10} u_{i+1/2}^{(0)} + \frac{3}{5} u_{i+1/2}^{(1)} + \frac{3}{10} u_{i+1/2}^{(2)}$$

would lead to a fifth order accurate linear scheme which is oscillatory.

3. Find the nonlinear weights w_0, w_1 and w_2 such that

$$u_{i+1/2} = w_0 u^{(0)}_{i+1/2} + w_1 u^{(1)}_{i+1/2} + w_2 u^{(2)}_{i+1/2}$$

   is both fifth order accurate in smooth regions **and** non-oscillatory for shocks. Thus we require the nonlinear weights w_0, w_1 and w_2 to satisfy the following two properties:

   - If $u(x)$ is smooth in all three substencils, then the nonlinear weights w_0, w_1 and w_2 are close to the linear weights γ_0, γ_1 and γ_2:

$$w_k = \gamma_k + O(\Delta x^2), \qquad k = 0, 1, 2.$$

   - If $u(x)$ has a discontinuity in the substencil S_k, then the corresponding w_k is very small:

$$w_k = O(\Delta x^4)$$

4. A robust choice of the nonlinear weights, given in Jiang and Shu (JCP96) and used in most WENO literature, is

$$w_k = \frac{\tilde{w}_k}{\tilde{w}_0 + \tilde{w}_1 + \tilde{w}_2}, \qquad \tilde{w}_k = \frac{\gamma_k}{(\varepsilon + \beta_k)^2}$$

where $\varepsilon = 10^{-6}$ typically (it can be adjusted by the average size of the solution), and the *smoothness indicator* β_k measures the smoothness of the function $u(x)$ in the substencil S_k and is given by

$$\beta_k = \Delta x_i \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (p_r'(x))^2 dx + \Delta x_i^3 \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (p_r''(x))^2 dx.$$

These smoothness indicators can be worked out explicitly as

$$
\begin{aligned}
\beta_0 &= \frac{13}{12}\left(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i\right)^2 + \frac{1}{4}\left(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i\right)^2 \\
\beta_1 &= \frac{13}{12}\left(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1}\right)^2 + \frac{1}{4}\left(\bar{u}_{i-1} - \bar{u}_{i+1}\right)^2 \\
\beta_2 &= \frac{13}{12}\left(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2 + \frac{1}{4}\left(3\bar{u}_i - 4\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2
\end{aligned}
$$

WENO finite volume schemes for conservation laws

A finite volume scheme approximates the conservation law

$$u_t + f(u)_x = 0$$

in its integral form

$$\frac{d}{dt}\bar{u}_i + \frac{1}{\Delta x_i}\left(f(u_{i+1/2}) - f(u_{i-1/2})\right) = 0 \tag{2}$$

To convert (2) to a finite volume scheme, we take our computational variables as the cell averages

$$\bar{u}_i, \qquad i = 1, 2, \cdots, N$$

and use the WENO reconstruction procedure described above to obtain an approximation to $u_{i+1/2}$. In practice, in order to obey upwinding for stability, we replace $f(u_{i+1/2})$ by

$$\hat{f}\left(u^-_{i+1/2}, u^+_{i+1/2}\right)$$

where $\hat{f}(u^-, u^+)$ is a monotone numerical flux satisfying

1. $\hat{f}(u^-, u^+)$ is non-decreasing in its first argument u^- and non-increasing in its second argument u^+, symbolically $\hat{f}(\uparrow, \downarrow)$;
2. $\hat{f}(u^-, u^+)$ is consistent with the physical flux $f(u)$, i.e. $\hat{f}(u, u) = f(u)$;
3. $\hat{f}(u^-, u^+)$ is Lipschitz continuous with respect to both arguments u^- and u^+.

Examples of monotone fluxes include

- The Godunov flux

$$\hat{f}^{God}(u^-, u^+) = \begin{cases} \min_{u^- \le u \le u^+} f(u), & \text{if } u^- \le u^+; \\ \max_{u^+ \le u \le u^-} f(u), & \text{if } u^- > u^+ \end{cases}$$

- The Lax-Friedrichs flux

$$\hat{f}^{LF}(u^-, u^+) = \frac{1}{2}\left(f(u^-) + f(u^+) - \alpha(u^+ - u^-)\right)$$

where $\alpha = \max_u |f'(u)|$

- The Engquist-Osher flux

$$\hat{f}^{LF}(u^-, u^+) = f^+(u^-) + f^-(u^+)$$

where

$$f^+(u) = f(0) + \int_0^u \max(f'(v), 0)dv;$$

$$f^-(u) = \int_0^u \min(f'(v), 0)dv$$

The approximations $u^-_{i+1/2}$ and $u^+_{i+1/2}$ are the WENO reconstructions from stencils one point biased to the left and one point biased to the right, respectively. For example, for a fifth order WENO scheme, the reconstruction $u^-_{i+1/2}$ uses the following 5 cell combined stencil

$$I_{i-2},\ I_{i-1},\ I_i,\ I_{i+1},\ I_{i+2}$$

and the reconstruction $u^+_{i+1/2}$ uses the following 5 cell combined stencil

$$I_{i-1},\ I_i,\ I_{i+1},\ I_{i+2},\ I_{i+3}.$$

Time discretization

The finite volume scheme described above can be written as a method-of-lines ODE

$$\begin{aligned}\frac{d}{dt}\bar{u}_i &= -\frac{1}{\Delta x_i}\left[\hat{f}\left(u^-_{i+1/2}, u^+_{i+1/2}\right) - \hat{f}\left(u^-_{i-1/2}, u^+_{i-1/2}\right)\right] \\ &\equiv L(\bar{u})_i\end{aligned}$$

This ODE system can be discretized in time by the TVD Runge-Kutta time discretization methods (Shu and Osher JCP87, also known as the SSP, or strong-stability-preserving methods, Gottlieb, Shu and Tadmor SIAM Rev 01).

For example, the third order version is:

$$
\begin{aligned}
\bar{u}^{(1)} &= \bar{u}^n + \Delta t L(\bar{u}^n) \\
\bar{u}^{(2)} &= \frac{3}{4}\bar{u}^n + \frac{1}{4}\bar{u}^{(1)} + \frac{1}{4}\Delta t L(\bar{u}^{(1)}) \\
\bar{u}^{n+1} &= \frac{1}{3}\bar{u}^n + \frac{2}{3}\bar{u}^{(2)} + \frac{2}{3}\Delta t L(\bar{u}^{(2)})
\end{aligned}
$$

WENO 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 $\hat{f}_{i+1/2} = \hat{f}\left(u_{i-p}, \cdots, u_{i+q}\right)$ 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 Δx 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

$$\frac{1}{\Delta x}\int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi)d\xi = f(u(x)) \tag{3}$$

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 (3).

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

$$\hat{f}_{i+1/2} = h(x_{i+1/2}) \tag{4}$$

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 (3) 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 (4).* Hence we can use the same reconstruction procedure discussed before for finite volume schemes!

Same subroutine, different input

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

and different output

$u^{\pm}_{i+1/2}$ 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

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

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

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
  - based on the cell averages $\{\bar{u}_i\}$, using integral form
  - need a reconstruction $\{\bar{u}_i\} \longrightarrow \{u^{\pm}_{i+1/2}\}$ (WENO)
  - can use any monotone flux $\hat{f}(u^-, u^+)$
  - does not need uniform or smooth meshes
  - larger computational cost for multi-dimensional problems

- Finite difference schemes
  - based on the point values $\{u_i\}$, using PDE form
  - need a reconstruction $\{f^{\pm}(u_i)\} \longrightarrow \{\hat{f}^{\pm}_{i+1/2}\}$ (WENO)
  - can only use monotone fluxes which correspond to smooth flux splitting $f(u) = f^+(u) + f^-(u)$
  - 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

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

where $\tilde{\bar{u}}$ is the cell average

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

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

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

where the numerical fluxes

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

$$\hat{g}_{i,j+1/2} \approx -\frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j+1/2}, t))dx \equiv \bar{g}_{i,j+1/2}$$

First, look at the simple, linear constant coefficient case

$$u_t + au_x + bu_y = 0$$

we would have

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

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

$$\{\tilde{\bar{u}}_{ij}\} \ \longrightarrow \ \{\tilde{u}_{i+1/2,j}\} \qquad \text{for fixed } i$$

$$\{\tilde{\bar{u}}_{ij}\} \ \longrightarrow \ \{\bar{u}_{i,j+1/2}\} \qquad \text{for fixed } j$$

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 $f(\tilde{u}) \neq \widetilde{f(u)}$, 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

$$\{\tilde{\bar{u}}_{ij}\} \longrightarrow \{\tilde{u}_{i+1/2,j}\} \longrightarrow \{u_{i+1/2,j+j_\alpha}\}_{\alpha=1}^{\alpha_g} \longrightarrow \{\hat{f}_{i+1/2,j}\}$$

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 at least 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 5 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.

Generalization to systems

For a hyperbolic system

$$u_t + f(u)_x = 0$$

Hyperbolicity means that $f'(u)$ (the Jacobian) has only real eigenvalues and a complete set of eigenvectors for all relevant u:

$$R^{-1}(u)\, f'(u)\, R(u) = \Lambda(u)$$

where

- $R(u)$: columns are the right eigenvectors of $f'(u)$;
- $R^{-1}(u)$: rows are the left eigenvectors of of $f'(u)$;
- $\Lambda(u)$: a diagonal matrix with the eigenvalues of $f'(u)$ on the diagonal

A local characteristic decomposition is usually used in the reconstruction step, to avoid the appearance of spurious oscillations when different waves from different characteristic fields interact. Thus, if the reconstruction of $u_{j+1/2}$ needs the cell averages $\bar{u}_{j-2}$, $\bar{u}_{j-1}$, $\bar{u}_j$, $\bar{u}_{j+1}$, $\bar{u}_{j+2}$, then the reconstruction proceeds as follows:

1. Find an approximation $\hat{u}_{j+1/2}$ which does not need to be accurate but should possess the following mean-value property if possible:

   $$f(\bar{u}_{j+1}) - f(\bar{u}_j) = f'(\hat{u}_{j+1/2})(\bar{u}_{j+1} - \bar{u}_j)$$

   Such $\hat{u}_{j+1/2}$ is called a Roe average of $\bar{u}_j$ and $\bar{u}_{j+1}$. If the Roe average does not exist or is difficult to compute, then a simple average

   $$\hat{u}_{j+1/2} = \frac{1}{2}(\bar{u}_j + \bar{u}_{j+1})$$

   is also fine.

2. Compute $R_{j+1/2} = R(\hat{u}_{j+1/2})$ and $R^{-1}_{j+1/2}$.

3. Project all the relevant cell averages to the local characteristic fields:

$$\bar{v}_{j+k} = R^{-1}_{j+1/2}\,\bar{u}_{j+k}, \qquad k = -2, \cdots, 2$$

4. Apply the WENO reconstruction procedure on each component of $\bar{v}_{j+k}$ to obtain $v_{j+1/2}$.

5. Obtain the reconstruction in the physical space by

$$u_{j+1/2} = R_{j+1/2}\,v_{j+1/2}$$

For multi-dimensional problems, the local characteristic decomposition procedure described above is applied in the normal direction of the cell edge where the numerical flux is computed.

Recent development and applications

Inverse Lax-Wendroff type boundary treatments

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.

For example, if we have the following scheme

$$u_j^{n+1} = au_{j-2}^n + bu_{j-1}^n + cu_j^n + du_{j+1}^n$$

with suitably chosen constants a, b, c and d (which depend on $\lambda = \frac{\Delta t}{\Delta x}$), approximating the PDE

$$u_t + u_x = 0,$$

$$u(x, 0) = f(x), \qquad u(0, t) = g(t)$$

to third order accuracy, then either a ghost point u_{-1}^n is needed, or the scheme cannot be used to compute u_1^{n+1}.

- 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.

- 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:

$$(u_t)_j = -(u_x)_j$$

$$(u_{tt})_j = -((u_x)_t)_j = -((u_t)_x)_j = (u_{xx})_j$$

...

- 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

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

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:

- 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:

$$u_x = -u_t; \qquad u_x(0,t) = -u_t(0,t) = -g'(t)$$

$$u_{xx} = (-u_t)_x = -(u_x)_t = u_{tt};$$

$$u_{xx}(0,t) = u_{tt}(0,t) = g''(t)$$

...

- Compute $g'(t)$, $g''(t)$, etc. either analytically or by finite difference.

Steady state Hamilton-Jacobi equations

We are interested in the steady state solution of the Hamilton-Jacobi equation

$$H(\phi_x, \phi_y) = f(x, y) \tag{5}$$

together with suitable boundary conditions.

We can use Runge-Kutta or other methods to march in time for the time dependent PDE

$$\phi_t + H(\phi_x, \phi_y) = f(x, y) \tag{6}$$

until steady state is reached, but that is rather slow.

One class of effective numerical methods is the fast sweeping method (Boué and Dupuis, SINUM 1999; Zhao, Math Comp 2005). For high order finite difference fast sweeping methods (Zhang, Zhao and Qian, JSC 2006), the first few points near an inflow boundary are usually prescribed to be the exact solution. This is not practical for problems with unknown exact solutions.

To fix the ideas, let us assume that the left boundary

$$\Gamma = \{(x, y) : x = 0,\ 0 \le y \le 1\} \tag{7}$$

of the computational domain $[0, 1]^2$ is the inflow boundary, on which the solution is given as

$$\phi(0, y) = g(y), \qquad 0 \le y \le 1.$$

We would like to obtain a high order approximation to the solution value $\phi_{i,j} \approx \phi(x_i, y_j)$ for $i = 1, 2$ and a fixed j, which corresponds to a point (x_i, y_j) near the inflow boundary which cannot be computed by the high order WENO scheme. A simple Taylor expansion gives, for $i = 1, 2$,

$$\phi(x_i, y_j) = \phi(0, y_j) + ih\,\phi_x(0, y_j) + \frac{(ih)^2}{2}\phi_{xx}(0, y_j) + O(h^3)$$

hence our desired approximation for the third order WENO scheme is

$$\phi_{i,j} = \phi(0, y_j) + ih\,\phi_x(0, y_j) + \frac{(ih)^2}{2}\phi_{xx}(0, y_j).$$

We already have $\phi(0, y_j) = g(y_j)$. The PDE (5), evaluated at the point $(0, y_j)$, becomes

$$H(\phi_x(0, y_j), g'(y_j)) = f(0, y_j) \tag{8}$$

in which the only unknown quantity is $\phi_x(0, y_j)$. Solving this (usually nonlinear) equation should give us $\phi_x(0, y_j)$. There might be more than one root, in which case we should choose the root so that

$$\partial_u H(\phi_x(0, y_j), g'(y_j)) > 0 \tag{9}$$

where ∂_u refers to the partial derivative with respect to the first argument in $H(u, v)$. The condition (9) guarantees that the boundary Γ in (7) is an inflow boundary. If the condition (9) still cannot pin down a root, then we would choose the root which is closest to the value from the first order fast sweeping solution at the same grid point.

To obtain $\phi_{xx}(0,y_j)$, we first take the derivative with respect to y on the original PDE (5), and then evaluate it at the the point $(0,y_j)$, which yields

$$\partial_u H(\phi_x(0,y_j),g'(y_j))\phi_{xy}(0,y_j)+\partial_v H(\phi_x(0,y_j),g'(y_j))g''(y_j) \\ = f_y(0,y_j), \tag{10}$$

where ∂_u and ∂_v refer to the partial derivatives with respect to the first and second arguments in $H(u,v)$, respectively. In this equation the only unknown quantity is $\phi_{xy}(0,y_j)$, hence we obtain easily its value, thanks to (9).

We then take the derivative with respect to x on the original PDE (5), and evaluate it at the the point $(0, y_j)$ to obtain

$$\partial_u H(\phi_x(0,y_j), g'(y_j))\phi_{xx}(0,y_j) + \partial_v H(\phi_x(0,y_j), g'(y_j))\phi_{xy}(0,y_j)$$
$$= f_x(0,y_j),$$

This time, the only unknown quantity is $\phi_{xx}(0,y_j)$, which we can obtain readily from this equality.

It is clear that this procedure can be carried out to any desired order of accuracy. Also, the inflow boundary Γ in (7) can be any piece of a smooth curve and does not need to be aligned with the mesh points: we only need to change the x and y partial derivatives to normal and tangential derivatives with respect to Γ. However, for this approach to work, Γ can not consist of a single point.

Example 1 (shape-from-shading). We solve the Eikonal equation

$$\sqrt{(\phi_x)^2 + (\phi_y)^2} = f(x, y)$$

with

$$\text{case (a):} \quad f(x, y) = \sqrt{(1 - |x|)^2 + (1 - |y|)^2}; \tag{11}$$

$$\text{case (b):} \quad f(x, y) = 2\sqrt{y^2(1 - x^2)^2 + x^2(1 - y^2)^2}. \tag{12}$$

The computational domain $\Omega = [-1, 1]^2$. The inflow boundary for this example is the whole boundary of the box $[-1, 1]^2$, namely $\Gamma = \{(x, y) : \; |x| = 1 \text{ or } |y| = 1\}$. The boundary condition $\phi(x, y) = 0$ is prescribed on Γ. We use third order WENO scheme with the fast sweeping method.

Table 1: Example 1. Lax-Wendroff type procedure for the inflow boundary. N is the number of mesh points in each direction.

N	L^1 error	L^∞	iter	L^1 error	L^∞	iter
	case (a)			case (b)		
80	2.06E-14	9.26E-13	1	8.64E-07	9.99E-04	35
160	1.58E-14	2.34E-12	1	5.25E-08	2.34E-04	47
320	1.09E-14	5.49E-12	1	4.43E-15	4.26E-12	62
640	1.01E-14	1.08E-11	2	1.41E-15	1.14E-12	90

Example 2. We solve the Eikonal equation with $f(x, y) = 1$. The computational domain is $[-1, 1]^2$, and the inflow boundary Γ is the unit circle of center (0,0) and radius 0.5, that is

$$\Gamma = \left\{ (x, y) : \ x^2 + y^2 = \frac{1}{4} \right\}.$$

The boundary condition $\phi(x, y) = 0$ is prescribed on Γ. The exact solution for this problem is the distance function to the circle Γ. This exact solution has a singularity at the center of the circle to which the characteristics converge, hence we exclude the box $[-0.15, 0.15]^2$ when measuring the errors. Notice that for this example, the domain boundary Γ is *not* aligned with the Cartesian mesh. We again use third order WENO scheme with the fast sweeping method.

Table 2: Example 2. Lax-Wendroff type procedure for the inflow boundary. N is the number of mesh points in each direction. The errors are measured in the computational domain but outside the box $[-0.15, 0.15]^2$.

N	L^1 error	order	L^∞	order	iteration number
80	0.573E-05		0.129E-03		25
160	0.122E-05	2.23	0.407E-05	4.98	32
320	0.191E-06	2.68	0.122E-05	1.74	46
640	0.246E-07	2.95	0.161E-06	2.92	62

References:

[1] L. Huang, C.-W. Shu and M. Zhang, *Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation*, Journal of Computational Mathematics, v26 (2008), pp.336-346.

[2] T. Xiong, M. Zhang, Y.-T. Zhang and C.-W. Shu, *Fifth order fast sweeping WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment*, Journal of Scientific Computing, v45 (2010), pp.514-536.

[3] Y.-T. Zhang, S. Chen, F. Li, H. Zhao and C.-W. Shu, *Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations*, SIAM Journal on Scientific Computing, v33 (2011), pp.1873-1896.

Time dependent conservation laws

We consider strongly hyperbolic conservation laws for $\boldsymbol{U} = \boldsymbol{U}(x, y, t) \in \mathbb{R}^2$

$$\begin{cases} \boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y = 0 & (x, y) \in \Omega, \quad t > 0, \\ \boldsymbol{U}(x, y, 0) = \boldsymbol{U}_0(x, y) & (x, y) \in \bar{\Omega}, \end{cases} \tag{13}$$

on a bounded domain Ω with appropriate boundary conditions prescribed on $\partial\Omega$ at time t. We assume Ω is covered by a uniform Cartesian mesh $\Omega_h = \{(x_i, y_j) : 0 \leqslant i \leqslant N_x, 0 \leqslant j \leqslant N_y\}$ with mesh size $\Delta x = \Delta y$.

To best illustrate the idea of the inverse Lax-Wendroff type procedure, we use 1D scalar conservation laws as an example

$$\begin{cases} u_t + f(u)_x = 0 & x \in (-1,1), \quad t > 0, \\ u(-1,t) = g(t) & t > 0, \\ u(x,0) = u_0(x) & x \in [-1,1]. \end{cases} \tag{14}$$

We assume $f'(u(-1,t)) \geqslant \alpha > 0$ and $f'(u(1,t)) \geqslant \alpha > 0$ 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

$$-1 + \Delta x/2 = x_0 < x_1 < \cdots < x_N = 1 - \Delta x/2. \qquad (15)$$

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

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

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

$$u_j \quad = \quad \sum_{k=0}^{s-1} \frac{(x_j+1)^k}{k!} \frac{\partial^k u}{\partial x^k}\bigg|_{x=-1, t=t_n}, \qquad j = -1, -2, -3. \quad (16)$$

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

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

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

$$u_{tt} + f''(u)u_t u_x + f'(u)u_{xt} \quad = \quad 0. \tag{17}$$

The term u_{xt} can be written as

$$\begin{aligned} u_{xt} &= (u_t)_x \\ &= -(f'(u)u_x)_x \\ &= -f''(u)u_x^2 - f'(u)u_{xx}. \end{aligned}$$

Substituting it into (17), we obtain an equation for u_{xx}

$$u_{tt} + f''(u)u_t u_x - f'(u)f''(u)u_x^2 - f'(u)^2 u_{xx} \quad = \quad 0. \tag{18}$$

Solving (18) for u_{xx} and evaluating it at $x = -1, t = t_n$, we have

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

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

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.

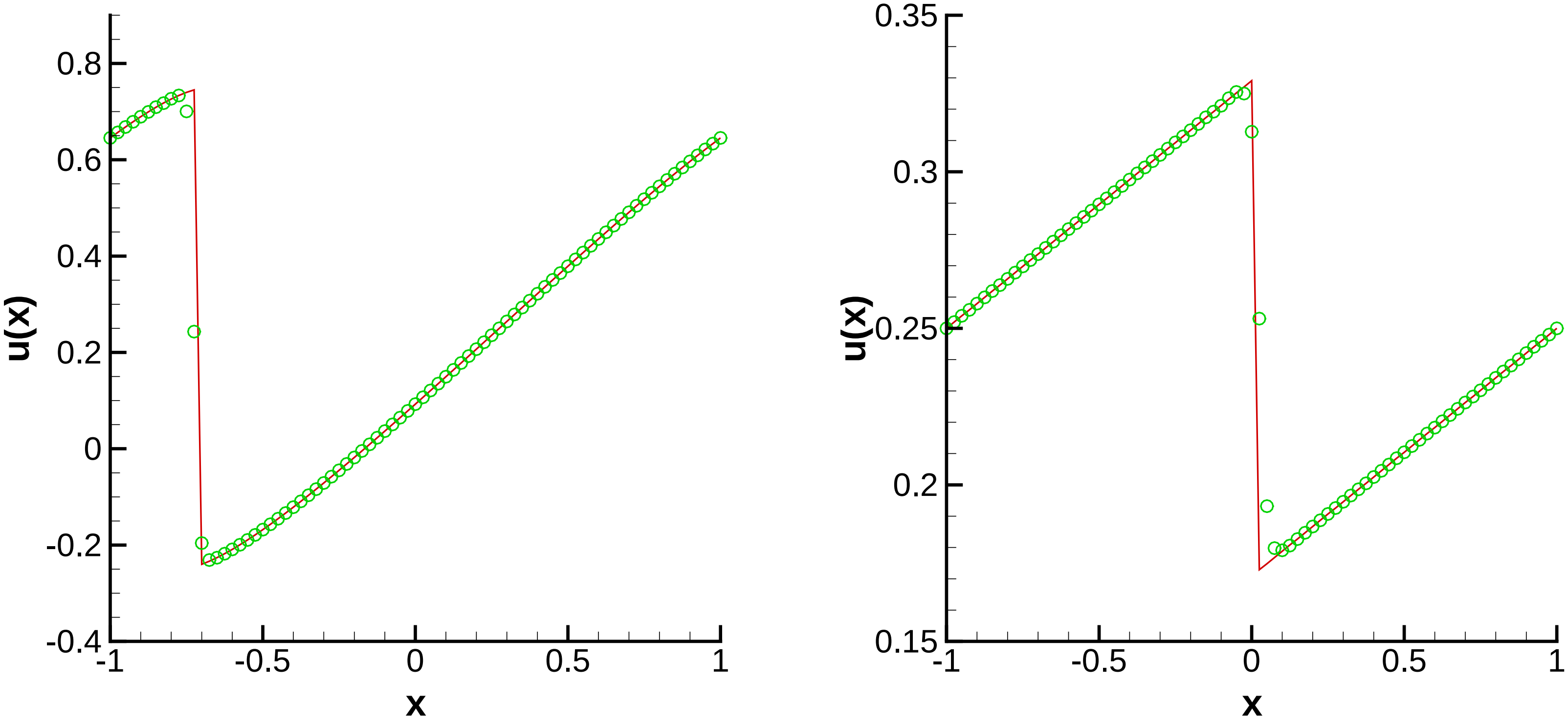
Example 3. We test the Burgers equation

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0 & x \in (-1,1), \quad t > 0, \\ u(x,0) = 0.25 + 0.5\sin(\pi x) & x \in [-1,1], \\ u(-1,t) = g(t) & t > 0. \end{cases} \tag{19}$$

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 3: Errors of the Burgers equation (19). $\Delta x = 2/N$ and $t = 0.3$.

N	L^1 error	order	L^∞ error	order
40	9.11E-05		3.56E-04	
80	3.10E-06	4.88	1.35E-05	4.72
160	1.31E-07	4.57	6.51E-07	4.38
320	3.97E-09	5.05	2.68E-08	4.60
640	1.02E-10	5.29	8.34E-10	5.00
1280	2.86E-12	5.15	2.62E-11	5.00



(a) $t = 1.1$

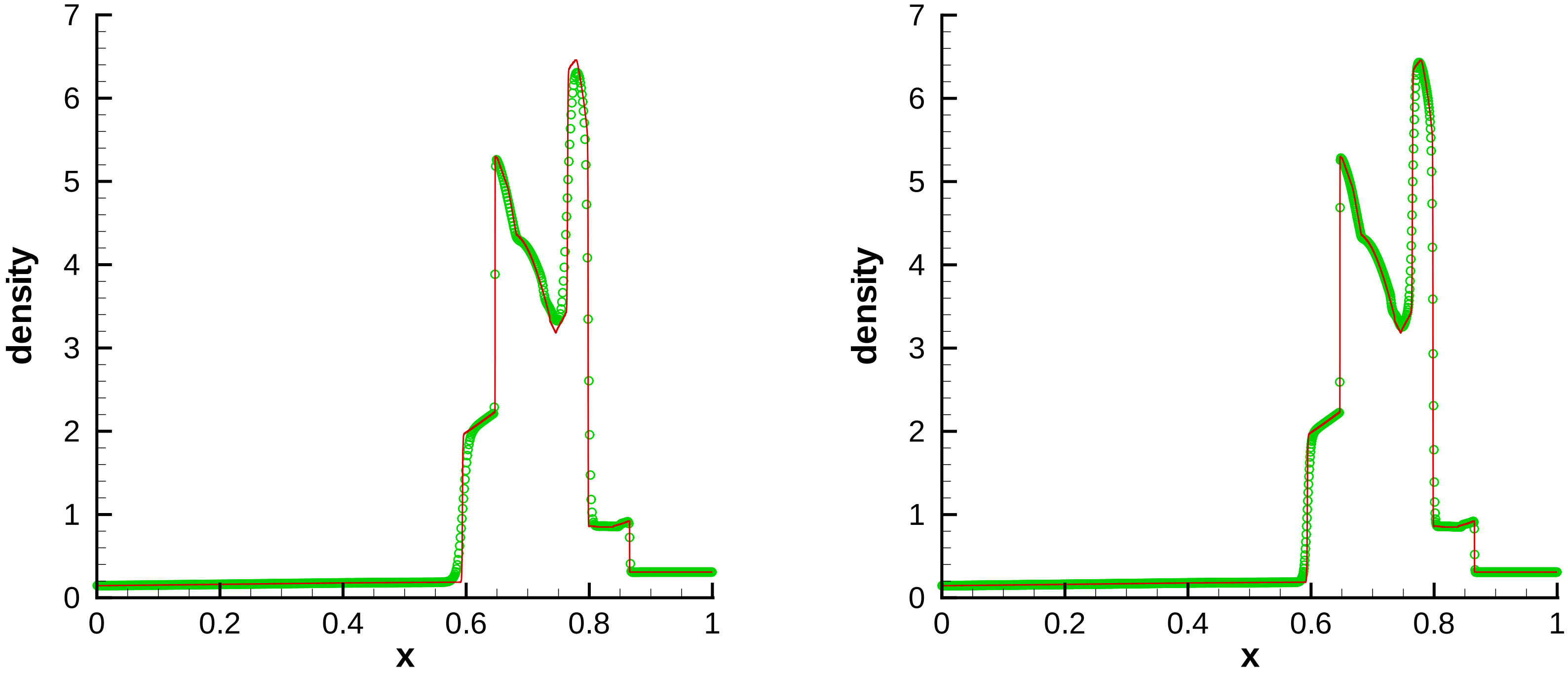
(b) $t = 12$

Figure 3: Burgers equation (19), $\Delta x = 1/40$. Solid line: exact solution; Symbols: numerical solution.

Example 4. Euler equations, blast wave example. We consider the interaction of two blast waves. The initial data are

$$\boldsymbol{U}(x,0) = \begin{cases} \boldsymbol{U}_L & 0 < x < 0.1, \\ \boldsymbol{U}_M & 0.1 < x < 0.9, \\ \boldsymbol{U}_R & 0.9 < x < 1, \end{cases}$$

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.



(a) $\Delta x = 1/800$

(b) $\Delta x = 1/1600$

Figure 4: 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 5. We test the 2D Burgers equation

$$\begin{cases} u_t + \frac{1}{2}\left(u^2\right)_x + \frac{1}{2}\left(u^2\right)_y = 0 & (x,y) \in \Omega, \quad t > 0, \\ u(x,y,0) = 0.75 + 0.5\sin\left[\pi(x+y)\right] & (x,y) \in \bar{\Omega}, \\ u(x,y,t) = g(x,y,t) & (x,y) \in \Gamma, \quad t > 0, \end{cases} \tag{20}$$

where

$$\begin{aligned} \Omega &= (-1,1) \times (-1,1), \\ \Gamma &= \{(x,y) : x = -1 \text{ or } y = -1\}, \end{aligned}$$

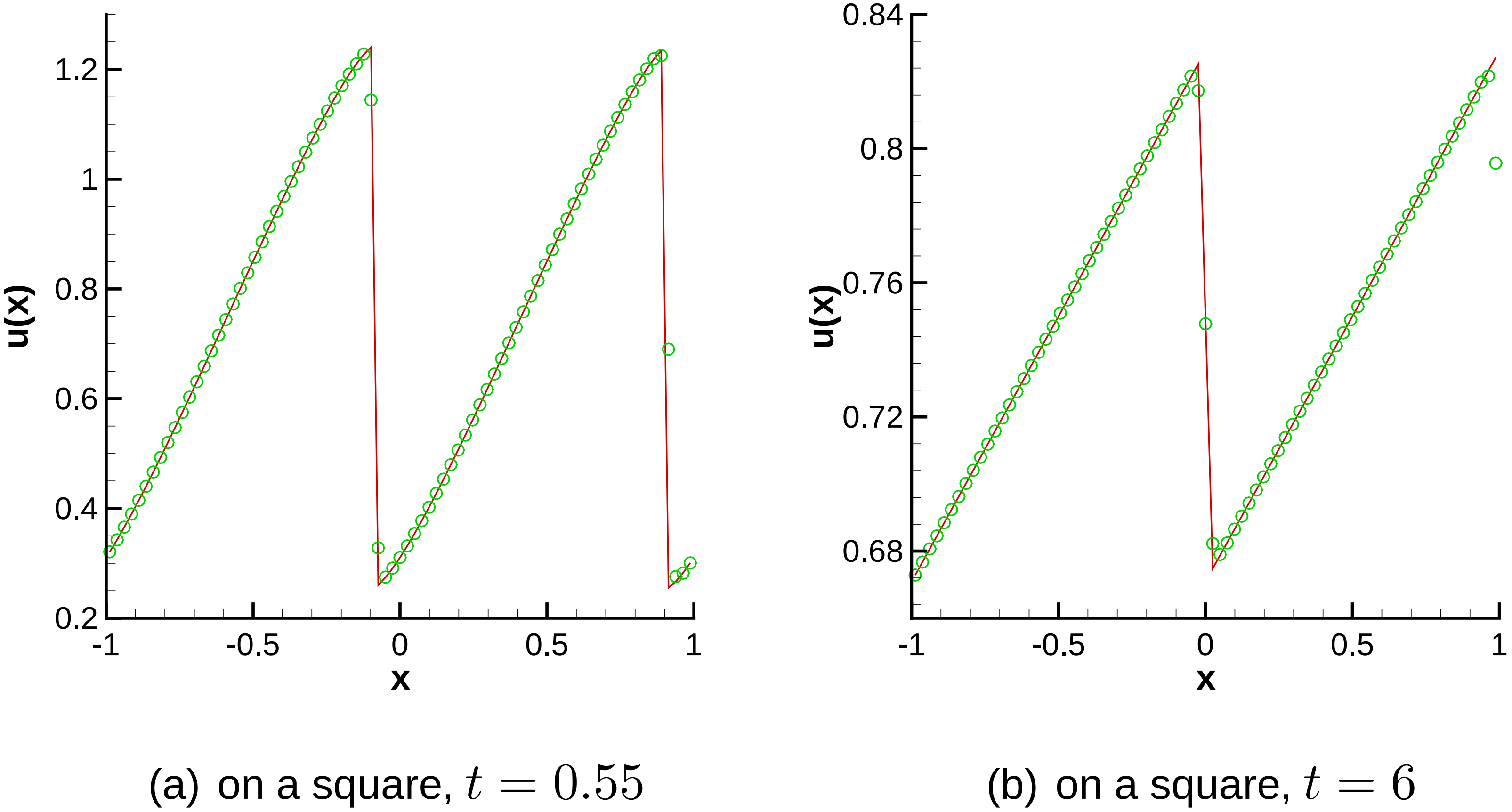
or

$$\begin{aligned} \Omega &= \{(x,y) : x^2 + y^2 < 0.5\}, \\ \Gamma &= \{(x,y) : x^2 + y^2 = 0.5 \text{ and } x + y \leqslant 0\}. \end{aligned}$$

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 4: Errors of the 2D Burgers equation (20). $\Delta x = 2/N_x, \Delta y = 2/N_y, t = 0.15$.

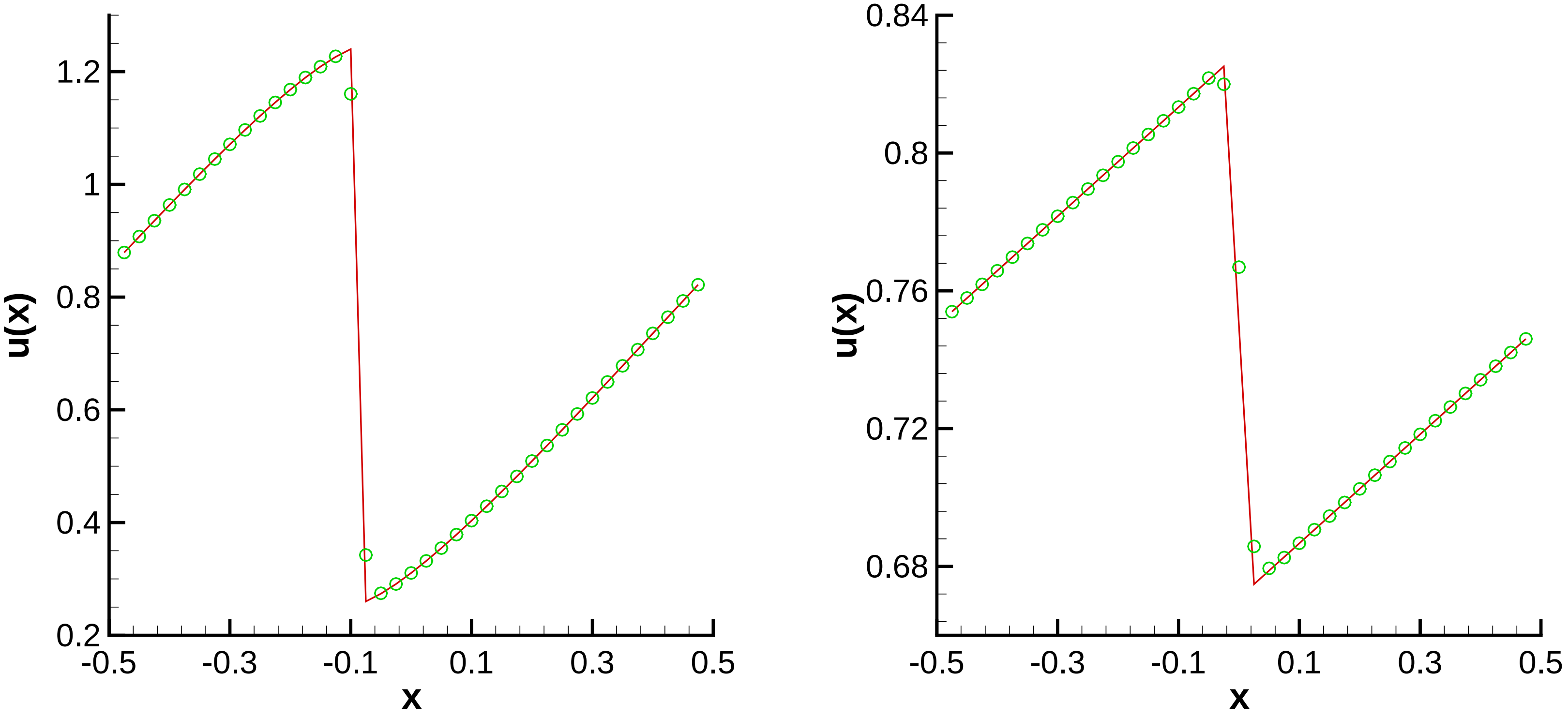
$N_x = N_y$	on a square				on a disk			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
40	1.55E-04		9.86E-03		1.10E-04		1.77E-03	
80	1.06E-05	3.87	1.80E-03	2.46	7.24E-06	3.93	4.06E-04	2.12
160	4.93E-07	4.43	2.38E-04	2.91	4.65E-07	3.96	4.77E-05	3.09
320	3.47E-08	3.83	2.83E-05	3.08	3.63E-08	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



(a) on a square, $t = 0.55$

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

Figure 5: 2D Burgers equation (20). $\Delta x = \Delta y = 1/40$. 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 6: Continued.

Example 6. For two-dimensional systems, we test the vortex evolution problem for the Euler equations. The mean flow is $\rho = p = u = v = 1$. We add to this mean flow an isentropic vortex perturbation centered at (x_0, y_0) in (u, v) and in the temperature $T = p/\rho$, no perturbation in the entropy $S = p/\rho^\gamma$

$$\begin{aligned}
(\delta u, \delta v) &= \frac{\epsilon}{2\pi} e^{0.5(1-r^2)} \left(-\bar{y}, \bar{x}\right), \\
\delta T &= -\frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2} e^{(1-r^2)}, \\
\delta S &= 0,
\end{aligned}$$

where $(\bar{x}, \bar{y}) = (x - x_0, y - y_0)$, $r^2 = \bar{x}^2 + \bar{y}^2$ and the vortex strength is $\epsilon = 5$.

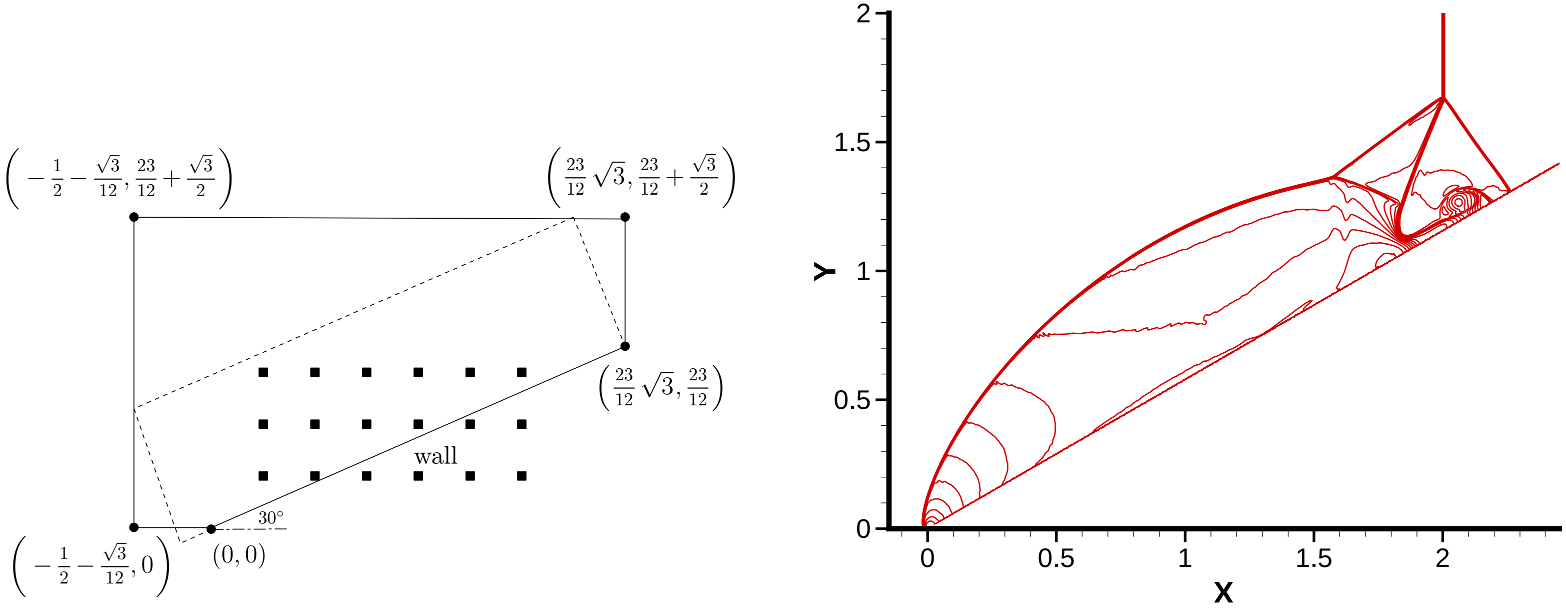
Table 5: Density errors and cost due to boundary treatment (in terms of ratio between the CPU time of boundary treatment and the total CPU time) of the vortex evolution problem on a square. The vortex is initially positioned at $(0,0)$. $\Delta x = 1.5/N$ and $t = 1$.

$N_x \times N_y$	L^1 error	order	L^∞ error	order	cost
40×40	3.41E-06		1.81E-05		0.56
80×80	2.38E-07	3.84	1.86E-06	3.28	0.40
160×160	2.08E-08	3.52	1.83E-07	3.34	0.25
320×320	2.40E-09	3.11	1.97E-08	3.22	0.14

Table 6: Density errors and cost due to boundary treatment (in terms of ratio between the CPU time of boundary treatment and the total CPU time) of the vortex evolution problem on a disk. The vortex is initially positioned at $(0.3, 0.3)$. $\Delta x = 2/N$ and $t = 0.1$.

$N_x \times N_y$	L^1 error	order	L^∞ error	order	cost
80×80	1.21E-07		2.29E-05		0.56
160×160	4.91E-09	4.62	1.29E-06	4.15	0.54
320×320	4.01E-10	3.61	1.27E-07	3.34	0.45
640×640	4.32E-11	3.21	1.60E-08	2.99	0.32

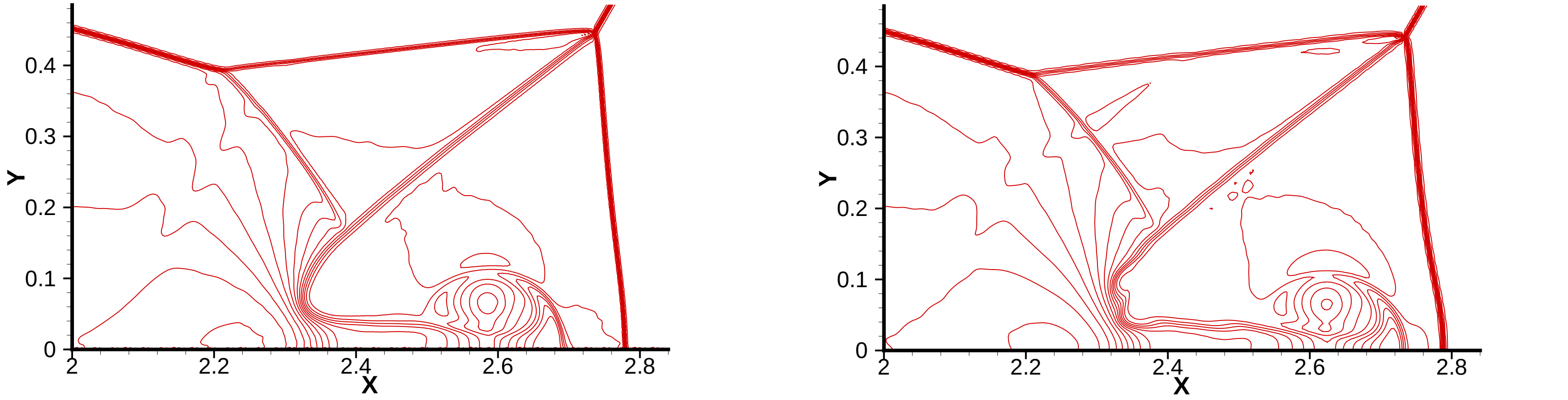
Example 7. We are most interested in applying our method to the solid wall boundary conditions $(u,v)\cdot\boldsymbol{n}=0$, when the wall is not aligned with the grid and can be curved. Our first example of this kind is the double Mach reflection problem. This problem is initialized by sending a horizontally moving shock into a wedge inclined by a 30° 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° 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.



(a) computational domain

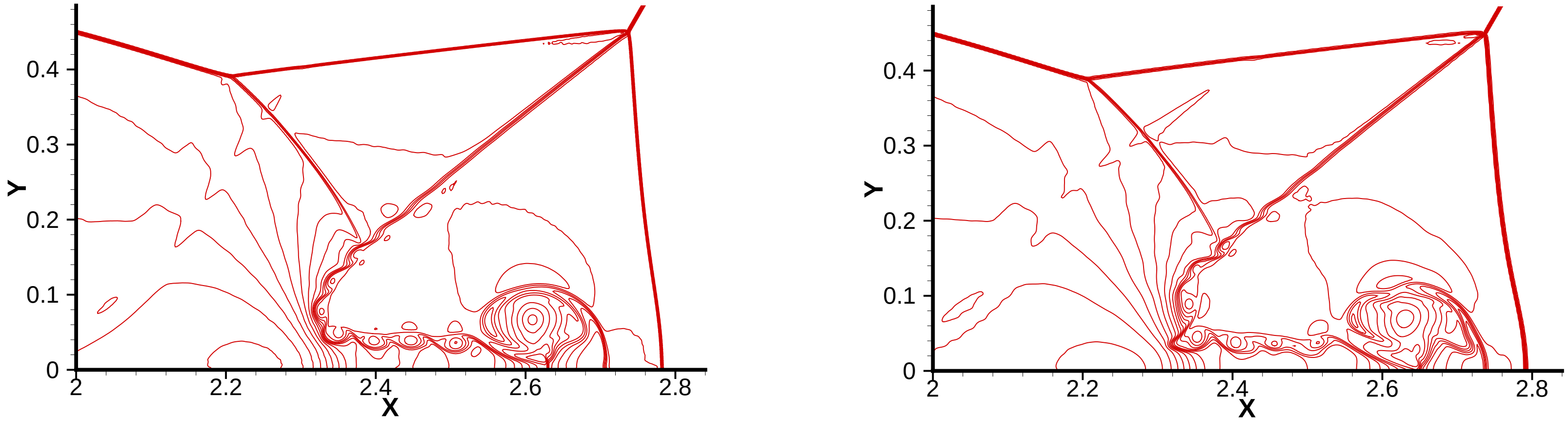
(b) density contour

Figure 7: 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. $\Delta x = \Delta y = \frac{1}{320}$.



(a) $\Delta x = \Delta y = \frac{1}{320}$, original problem (b) $\Delta x = \Delta y = \frac{\sqrt{3}}{480}$, equivalent problem

Figure 8: 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 9: Continued

.

Example 8. 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.

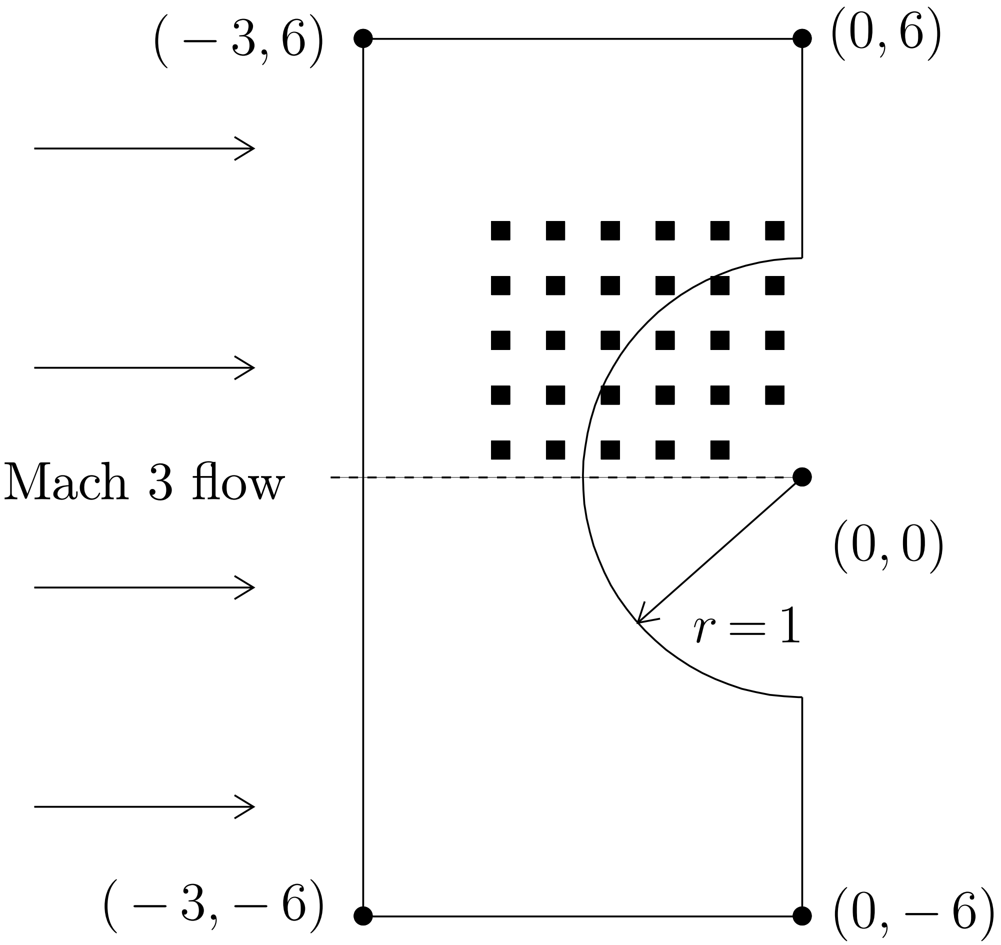
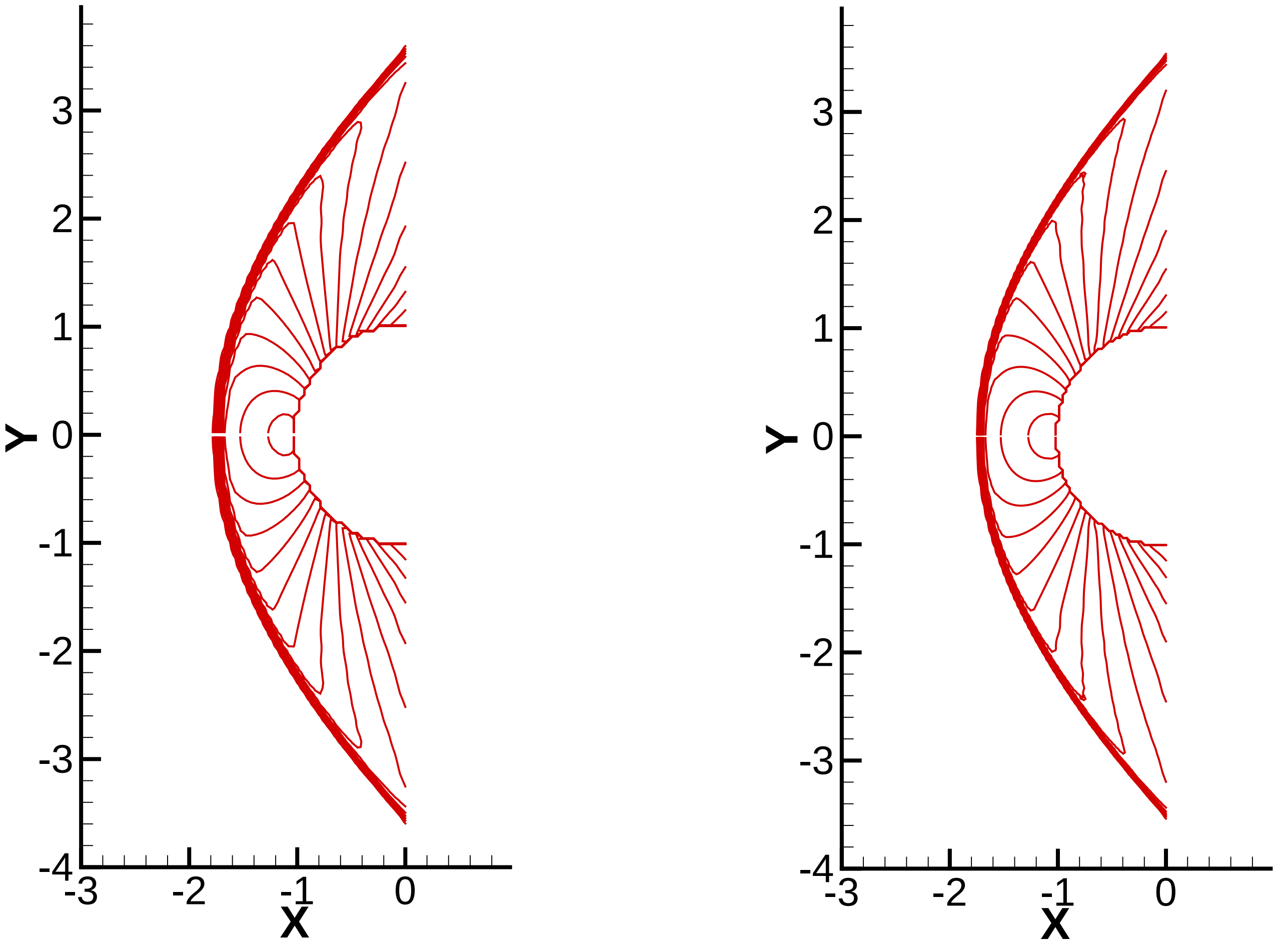


Figure 10: Physical domain of flow past a cylinder. The square points indicate some of the grid points near the cylinder. Illustrative sketch, not to scale.



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

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

Figure 11: Pressure contour of flow past a cylinder.

Reference:

[4] S. Tan and C.-W. Shu, *Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws*, Journal of Computational Physics, v229 (2010), pp.8144-8166.

[5] S. Tan, C. Wang, C.-W. Shu and J. Ning, *Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws*, Journal of Computational Physics, v231 (2012), pp.2510-2527.

Compressible inviscid flows involving complex moving geometries

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.
- 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 moving Lagrangian framework.

Example 9. 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

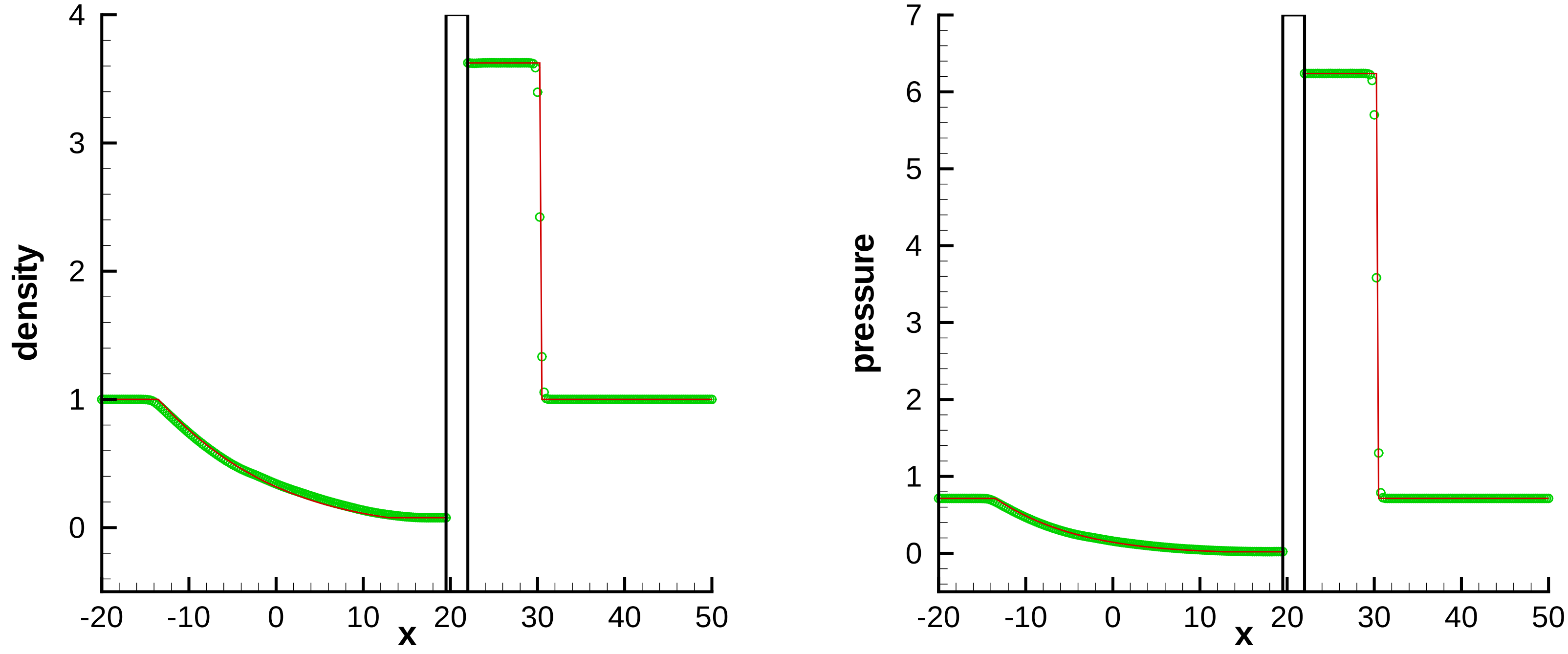
$$\begin{aligned}
\rho(x,0) &= 1 + 0.2\cos\left[2\pi\left(x-0.5\right)\right],\\
u(x,0) &= x-1,\\
p(x,0) &= \rho(x,0)^\gamma,
\end{aligned}$$

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 7: Entropy errors and convergence rates of Example 9

h	$x_l(t) = 0.5(1 - \sin t)$			
	L^1 error	order	L^∞ error	order
1/40	7.26E-07		1.32E-06	
1/80	1.15E-08	5.98	2.82E-08	5.55
1/160	3.43E-10	5.07	6.19E-10	5.51
1/320	9.90E-12	5.11	2.49E-11	4.64

Example 10. 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.



(a) Density

(b) Pressure

Figure 12: Density and pressure profiles of Example 10. The piston is represented by the rectangle. Solid lines: exact solutions; Symbols: numerical solutions with $h = 0.25$.

Example 11. We now move on to 2D examples. We first test a 2D version of Example 9. A gas is confined in a rectangular region whose boundaries are rigid walls. The top and bottom walls are fixed at $y = 0$ and $y = 1$ respectively. The right wall is fixed at $x = 1$. The left moving wall is positioned at $x_l(t) = 0.5(1 - \sin t)$. The initial conditions are

$$\begin{aligned}
\rho(x,y,0) &= 1 + 0.2\cos\left[2\pi\left(x - 0.5\right)\right] + 0.1\cos\left[2\pi(y - 0.5)\right],\\
u(x,y,0) &= x - 1,\\
v(x,y,0) &= y(1-y)\cos(\pi x),\\
p(x,y,0) &= \rho(x,y,0)^{\gamma},
\end{aligned}$$

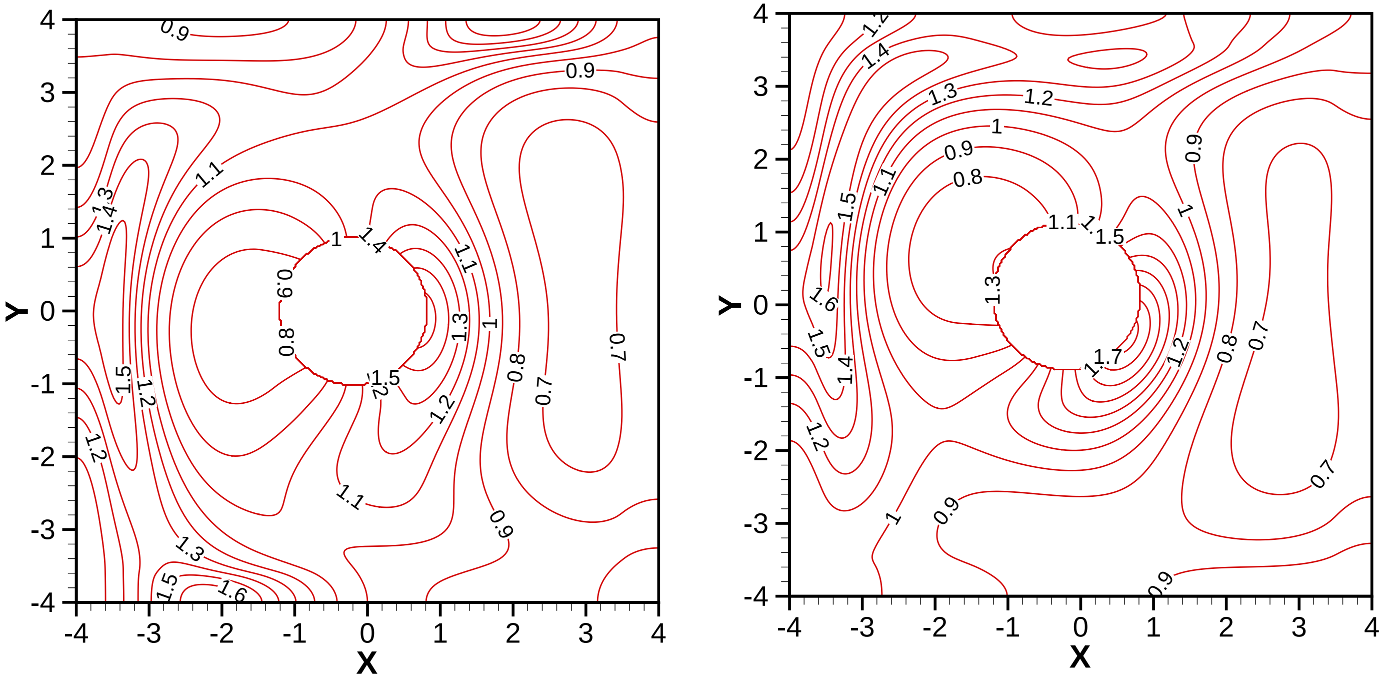
such that the initial entropy $s(x, y, 0) = 1$. We use our high order boundary treatment at the left moving wall and the reflection technique at the fixed walls.

Table 8: Entropy errors and convergence rates of Example 11.

h	L^1 error	order	L^∞ error	order
1/80	2.50E-08		3.28E-07	
1/160	1.10E-09	4.50	3.06E-08	3.42
1/320	9.70E-11	3.50	6.17E-09	2.31
1/640	9.87E-12	3.30	7.06E-10	3.13

Example 12. 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 $\boldsymbol{X}_c(t)$. 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 $\boldsymbol{X}_c = (-0.5\sin t, 0)$ such that the cylinder moves horizontally. In the second case, we take $\boldsymbol{X}_c = (-0.5\sin t, 0.3t)$ such that the cylinder moves in the 2D space.



(a) $\boldsymbol{X}_c = (-0.5\sin t, 0)$

(b) $\boldsymbol{X}_c = (-0.5\sin t, 0.3t)$

Figure 13: Density contours of Example 12. $h = 1/40$, $t = 0.4$.

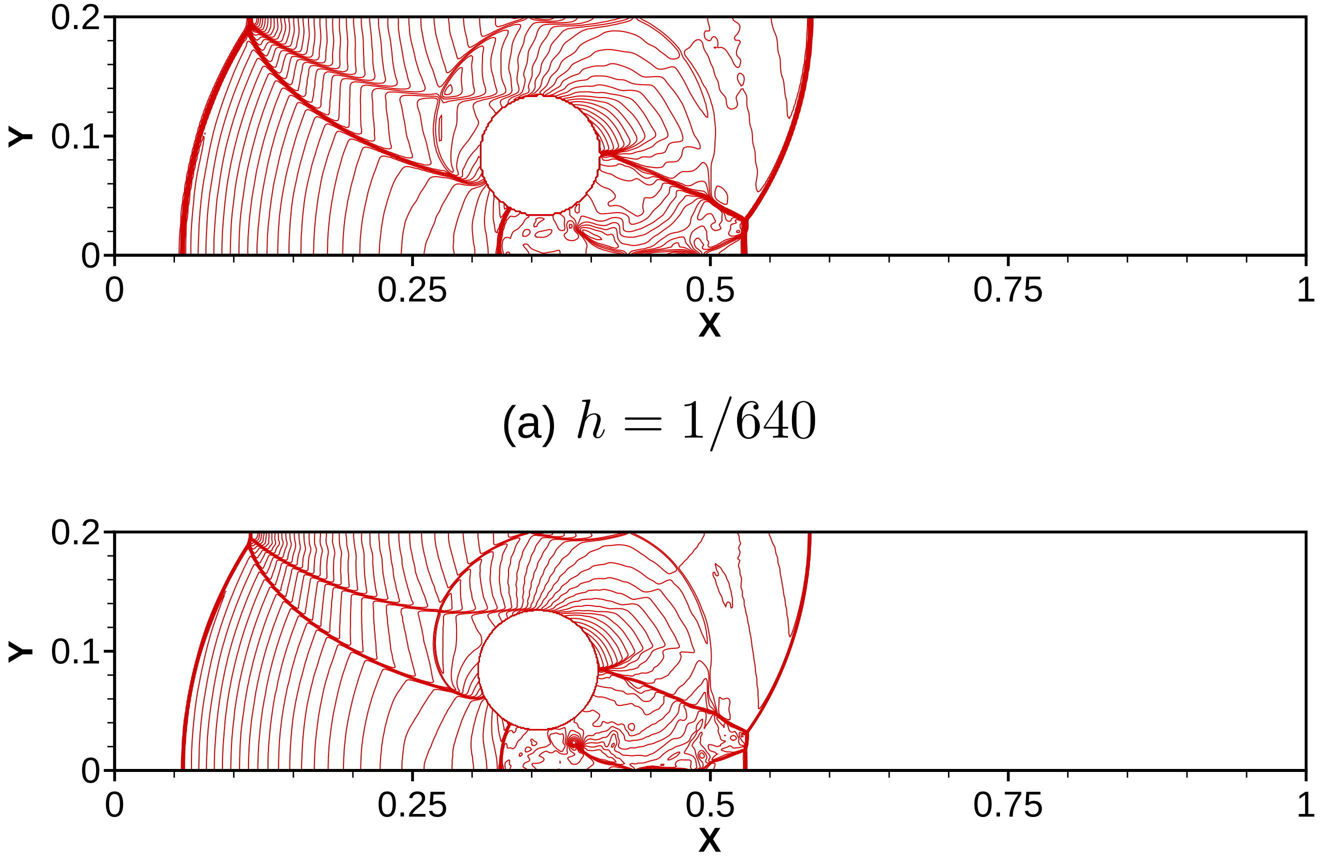
Table 9: Entropy errors and convergence rates of Example 12. $t = 0.4$.

h	$\mathbf{x}_c = (-0.5\sin t, 0), t = 0.7$				$\mathbf{x}_c = (-0.5\sin t, 0.3t), t = 0.5$			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
1/5	4.11E-03		2.47E-03		3.93E-03		1.79E-03	
1/10	3.86E-04	3.41	3.60E-04	2.78	4.05E-04	3.28	1.93E-04	3.21
1/20	1.21E-05	5.00	1.12E-05	5.00	1.20E-05	5.07	9.20E-06	4.39
1/40	2.43E-07	5.64	6.08E-07	4.21	2.21E-07	5.77	3.73E-07	4.62

Example 13. 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 10: Center of the cylinder of Example 13.

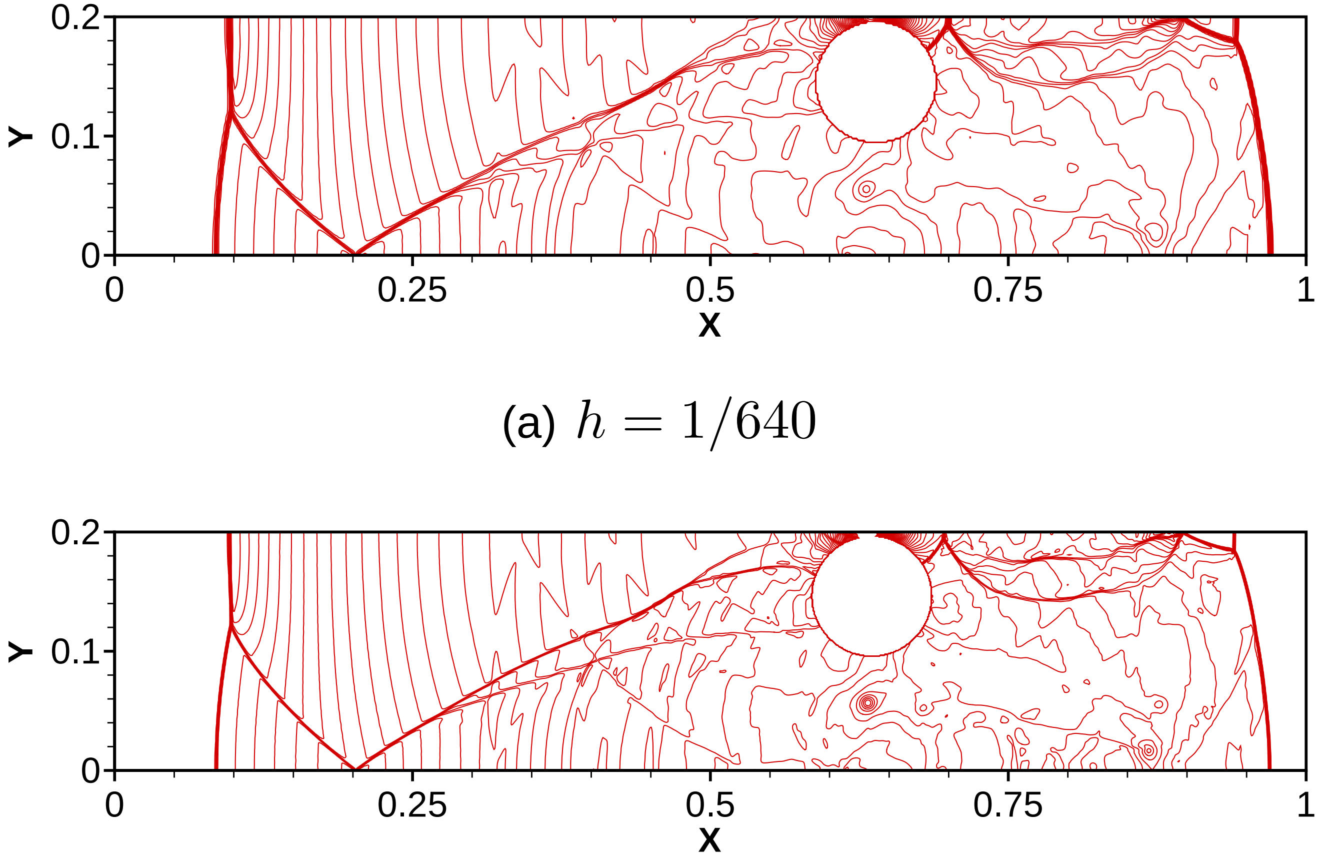
h	$t = 0.1641$		$t = 0.30085$	
	x-coordinate	y-coordinate	x-coordinate	y-coordinate
1/160	3.7058E-01	8.1140E-02	6.7178E-01	1.3759E-01
1/320	3.6153E-01	8.3219E-02	6.4959E-01	1.4444E-01
1/640	3.5706E-01	8.3680E-02	6.3895E-01	1.4517E-01
1/1280	3.5539E-01	8.4133E-02	6.3550E-01	1.4607E-01
1/2560	3.5461E-01	8.4258E-02	6.3362E-01	1.4638E-01



(a) $h = 1/640$

(b) $h = 1/1280$

Figure 14: Pressure contours at $t = 0.1641$. 53 contours from 2 to 28. $t = 0.1641$.



(a) $h = 1/640$

(b) $h = 1/1280$

Figure 15: Pressure contours at $t = 0.30085$. 53 contours from 2 to 28.

$t = 0.30085$.

Reference:

[6] S. Tan and C.-W. Shu, *A high order moving boundary treatment for compressible inviscid flows*, Journal of Computational Physics, v230 (2011), pp.6023-6036.

[7] S. Tan and C.-W. Shu, *Inverse Lax-Wendroff procedure for numerical boundary conditions of hyperbolic equations: survey and new developments*, in “Advances in Applied Mathematics, Modeling and Computational Science”, R. Melnik and I. Kotsireas, Editors, Fields Institute Communications 66, Springer, New York, 2013, pp.41-63.

Concluding remarks of the inverse LW procedure

- We have demonstrated an inverse Lax-Wendroff procedure for boundary treatment, which yields stable discretization with the same CFL number as the inner scheme and allows us to compute problems on arbitrary domains using Cartesian meshes.
- The technique can be applied to inviscid flows with complex moving geometries, yielding stable and high order accurate solutions.
- Future work would involve a generalization of this technique to other schemes such as the discontinuous Galerkin method, and to viscous problems and to problems with deformable structures.

Free-stream preserving finite difference schemes on curvilinear meshes

An important property for finite difference schemes designed on curvilinear meshes is the exact preservation of free-stream solutions.

This property is difficult to fulfill for high order conservative essentially non-oscillatory (WENO) finite difference schemes, mainly because the standard finite difference WENO reconstruction is performed directly on the flux values $\{f(u_i)\}$ (or $\{f^+(u_i)\}$ and $\{f^-(u_i)\}$ with a flux splitting $f(u) = f^+(u) + f^-(u)$), not on the point values of the solution $\{u_i\}$.

The following one-dimensional Euler equations with uniform grids are used for the discussion of the scheme.

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = 0, \tag{21}$$

where

$$\begin{aligned} Q &= (\rho, \rho u, e)^T, \\ F &= (\rho u, \rho u^2 + p, u(e+p))^T. \end{aligned} \tag{22}$$

On a uniform mesh $x_i = i\Delta x$, we would like to find a consistent numerical flux function

$$\hat{F}_{i+\frac{1}{2}} = \hat{F}(Q_{i-r}, \ldots, Q_{i+s}), \tag{23}$$

such that the flux difference approximates the derivative $F(Q(x))_x$ to k-th order accuracy

$$\frac{1}{\Delta x}(\hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}}) = F(Q(x))_x|_{x_i} + O(\Delta x^k). \qquad (24)$$

The alternative flux formulation, first developed in Shu and Osher JCP88 and extensively explored in Jiang, Shu and Zhang SISC13 in the context of WENO interpolation, is given as follows

$$\hat{F}_{i+\frac{1}{2}} = F_{i+\frac{1}{2}} + \sum_{\ell=1}^{[(r-1)/2]} a_{2\ell} \Delta x^{2\ell} \left(\frac{\partial^{2\ell}}{\partial x^{2\ell}} F \right)_{i+\frac{1}{2}} + O(\Delta x^{r+1}), \qquad (25)$$

which guarantees $k = r$-th order accuracy in (24).

The coefficients $a_{2\ell}$ in (25) can be obtained through Taylor expansion and the accuracy constraint (24). To get an approximation with fifth order accuracy ($k = 5$ in (24), thus $r = 4$ in (25)), we have

$$\hat{F}_{i+\frac{1}{2}} = F_{i+\frac{1}{2}} - \frac{1}{24}\Delta x^2 \frac{\partial^2 F}{\partial x^2}|_{i+\frac{1}{2}} + \frac{7}{5760}\Delta x^4 \frac{\partial^4 F}{\partial x^4}|_{i+\frac{1}{2}}. \qquad (26)$$

The first term of the numerical flux in (26) is approximated by

$$F_{i+\frac{1}{2}} = h(Q^-_{i+\frac{1}{2}}, Q^+_{i+\frac{1}{2}}) \qquad (27)$$

with the values $Q^{\pm}_{i+\frac{1}{2}}$ obtained by a WENO interpolation based on neighboring point values Q_j using the local characteristic variables. The two-argument numerical function h is based on an exact or approximate Riemann solver. For example, we can use the Godunov flux, the Lax-Friedrichs flux, the HLLC flux, etc.

The remaining terms of the numerical flux in (26) or (25) have at least Δx^2 in their coefficients, hence they only need lower order approximations and they are expected to contribute much less to spurious oscillations. It is the conclusion of Jiang, Shu and Zhang SISC13 that these remaining terms can be approximated by simple central approximation or one-point upwind-biased approximation with suitable orders of accuracy, without using the more expensive WENO procedure.

Using this formulation and a careful design of metric derivative approximations, Jiang, Shu and Zhang (MAA to appear) designed a high order finite difference WENO scheme on 3D curvilinear coordinates which preserves exactly free-stream solutions.

References:

Y. Jiang, C.-W. Shu and M. Zhang, *An alternative formulation of finite difference weighted ENO schemes with Lax-Wendroff time discretization for conservation laws*, SIAM Journal on Scientific Computing, v35 (2013), pp.A1137-A1160.

Y. Jiang, C.-W. Shu and M. Zhang, *Free-stream preserving finite difference schemes on curvilinear meshes*, Methods and Applications of Analysis, to appear.

A homotopy method for solving steady state problems

When a WENO finite difference scheme is used to approximate a steady state problem, the resulting nonlinear algebraic system is very difficult to solve.

One can follow physical time and march to steady state, but this could be very slow, unless a good preconditioner is used.

Alternatively, one could use an iterative method to solve this nonlinear algebraic system directly, for example some kind of Newton's method. However, attention must be paid to obtain the correct entropy solution which is stable under time marching.

Homotopy continuation is an efficient tool for solving polynomial systems. Its efficiency relies on utilizing adaptive stepsize and adaptive precision path tracking, and endgames.

Consider the following one-dimensional hyperbolic conservation laws

$$u_t + \big(f(u)\big)_x = g(u,x).$$

Setting u_t to zero, the steady state problem becomes

$$\big(f(u)\big)_x - g(u,x) = 0.$$

For an initial condition u^0, we introduce the homotopy

$$H(u,\epsilon) = \left(\big(f(u)\big)_x - g(u,x) - \epsilon u_{xx}\right)(1-\epsilon)+\epsilon(u-u^0) \equiv 0, \quad (28)$$

where ϵ is a parameter between 0 and 1. In particular, when $\epsilon = 1$, the initial condition automatically satisfies (28) and, when $\epsilon = 0$, (28) becomes the steady state problem. The term ϵu_{xx} is introduced and guarantees that steady states obtained by homotopy function satisfy the entropy condition. It can also smooth the solution during the iterations.

Consider the steady state solutions of the Burgers equation with a source term

$$u_t + \left(\frac{u^2}{2}\right)_x \;=\; \sin(x)\cos(x), \;\; x \in [0, \pi]$$

with initial condition $u(x,0) = \beta \sin(x)$ and boundary condition $u(0,t) = u(\pi,t) = 0$. The steady state solution to this problem depends upon the value of β: a shock forms within the domain if $\beta \in [-1, 1]$; otherwise, the steady state solution is smooth.

We take $\beta = 2$ yielding $u(x, \infty) = \sin x$. We use our homotopy method with the Lax-Friedrichs WENO3 fluxes. The convergence to third order accuracy of L^1 and L^∞ error is clearly observed from this data. It is obvious that the homotopy method is much more efficient than the time marching approach. Furthermore, via a mesh refinement study, we observe that the CPU time increases linearly for the homotopy method. This shows that the computational cost is $O(N)$ (N is the number of spatial grid points) and the homotopy approach is free of the CFL condition.

Table 11: Errors and numerical orders of accuracy of WENO3 scheme for Burgers example with N points

N	L^1 error	Order	L^∞ error	Order	computing time	
					homotopy	TVD-RK2
20	3.68e-2	–	1.55e-2	–	0.21s	0.18s
40	7.49e-3	2.30	4.38e-2	1.83	0.47s	0.69s
80	1.21e-3	2.63	9.12e-3	2.26	1.02s	2.52s
160	1.71e-4	2.82	1.60e-3	2.51	1.98s	9.85s
320	2.18e-5	2.97	2.24e-4	2.84	4.03s	39.10s
640	2.76e-6	2.98	2.90e-5	2.95	8.48s	154.93s

Our last example is to compute a steady state solution of a regular shock reflection problem for the two-dimensional Euler equations.

The numerical solutions obtained using the homotopy method with the WENO third order scheme are displayed in Figure 16. It can be clearly seen that the incident and reflected shocks are well-resolved. Figure 17 shows the convergence of the solution. It takes 22 steps (489.23 seconds) while using the Newton iteration based method takes 46 steps (1,012.37 seconds).

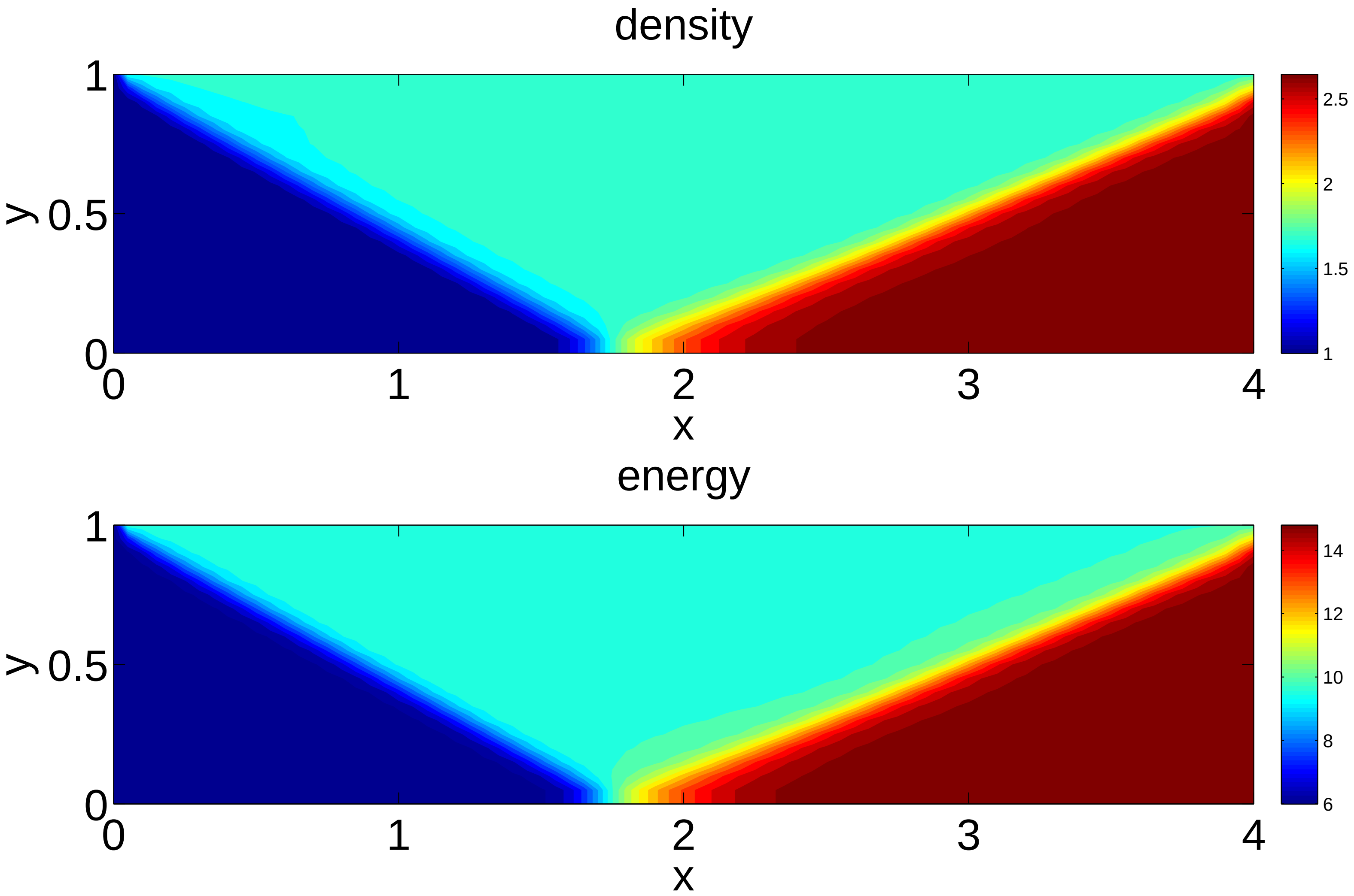


Figure 16: Shock reflection for the density and the energy respectively with 100×25 grid points.

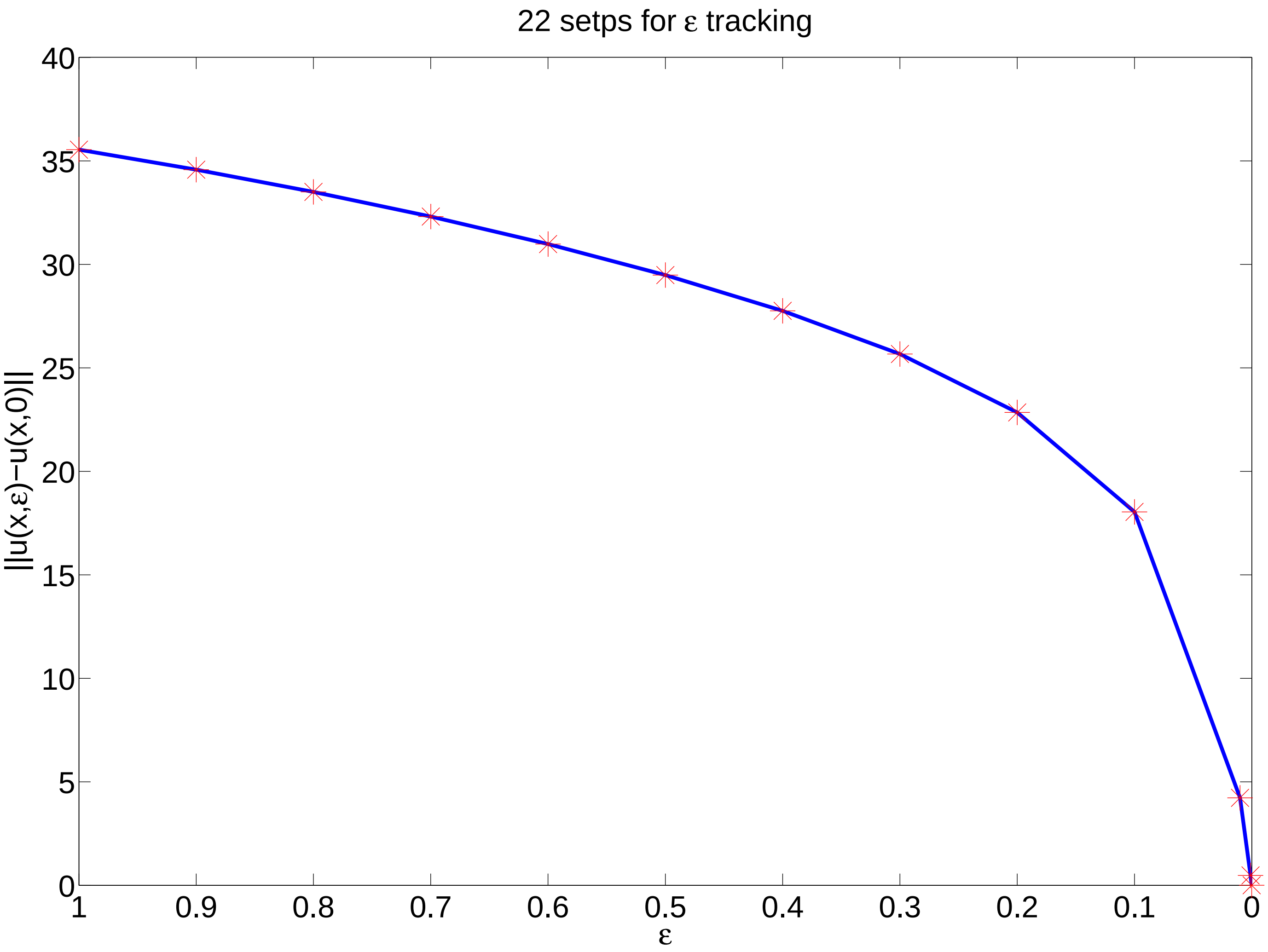


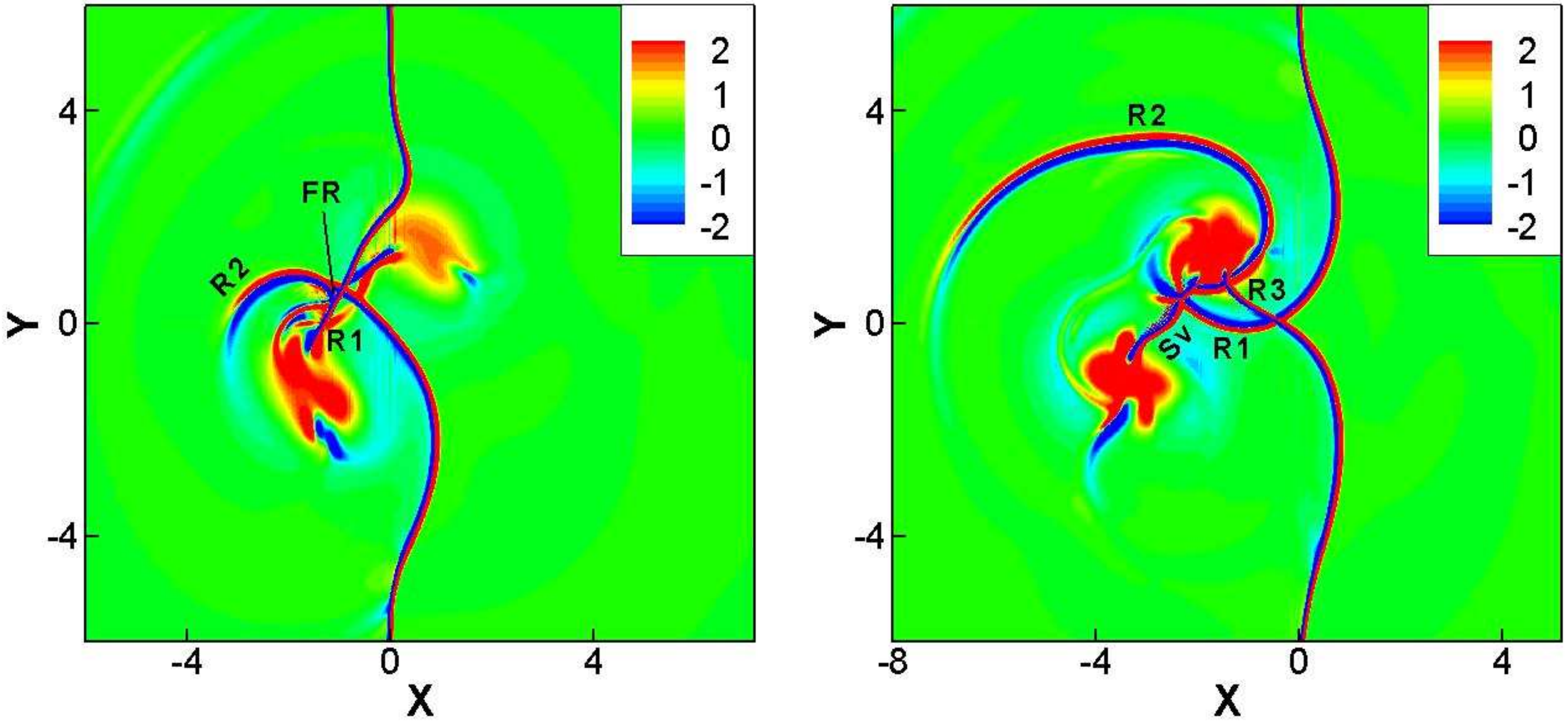
Figure 17: Convergence of shock reflection example. The maximum stepsize is $1/10$.

Reference:

W. Hao, J.D. Hauenstein, C.-W. Shu, A.J. Sommese, Z. Xu and Y.-T. Zhang, *A homotopy method based on WENO schemes for solving steady state problems of hyperbolic conservation laws*, Journal of Computational Physics, v250 (2013), pp.332-346.

Application: Shock-vortex and vortex-vortex interactions

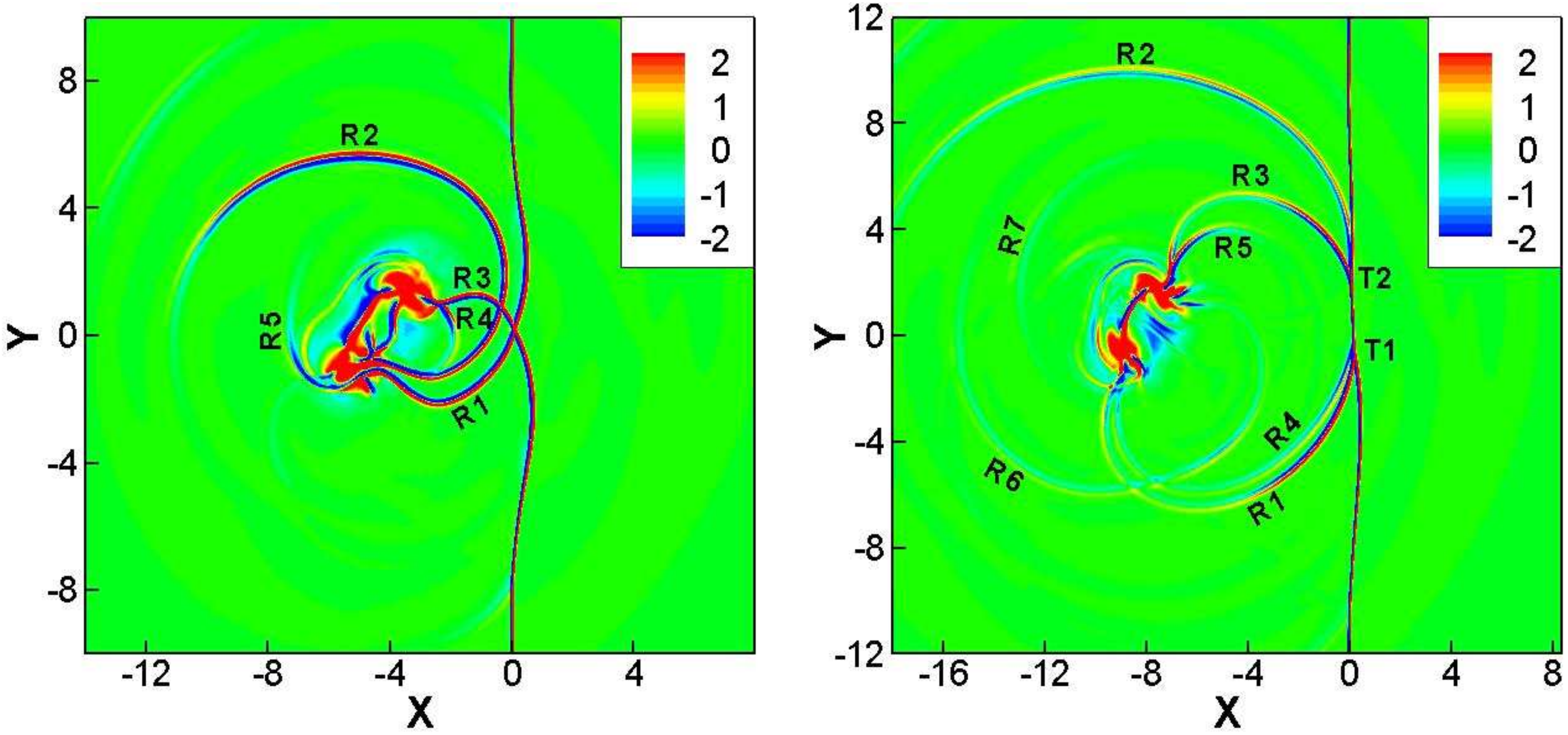
An example where the solution contains both discontinuities and rich structures in smooth regions: This is the problem of a strong shock interacting with a pair of vortices (Zhang, Zhang and Shu, Physics of Fluids 2006), simulated by the fifth order finite difference WENO scheme on the compressible Navier-Stokes equations. In Figure 18 we plot the shadowgraphs (contours of $\nabla^2\rho$ where ρ is the density) of an oblique Mach 1.2 shock with a strong colliding vortex pair. We can see clearly that complicated flow structure from the shock vortex interaction is resolved well by the fifth order WENO scheme.



(a) $t = 5.0$

(b) $t = 7.0$

Figure 18: Fifth order WENO simulation of the evolution of an oblique shock and a colliding vortex pair interaction. Shock Mach number $M_s = 1.2$, vortex Mach number $M_\nu = 0.8$, and angle of the oblique shock wave $\alpha = 45°$.

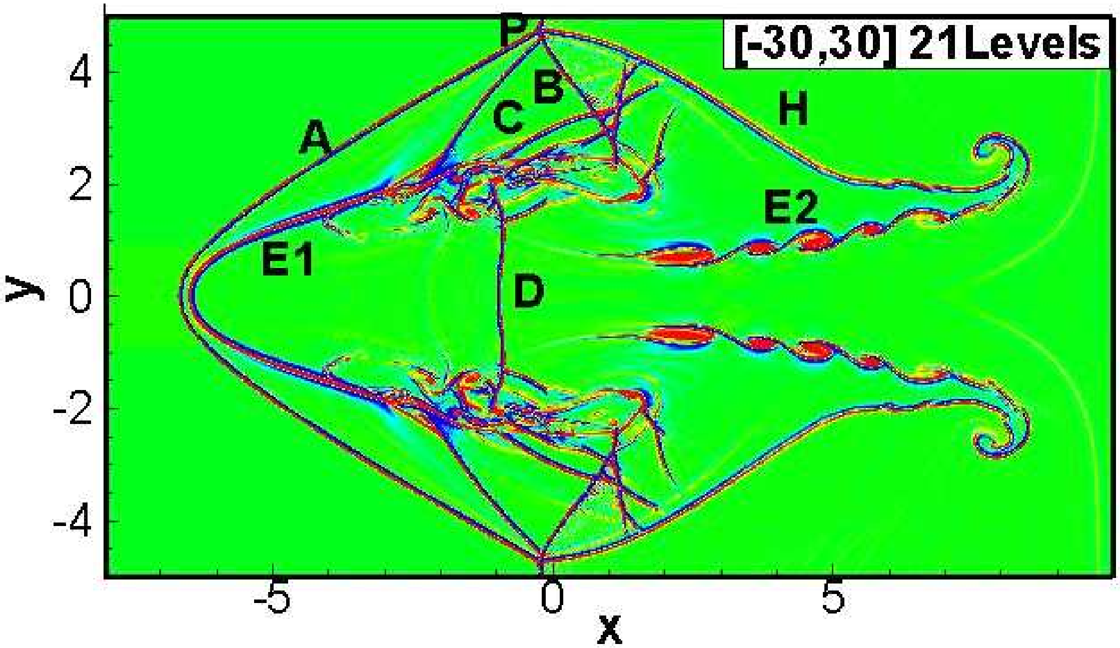


(c) $t = 9.0$

(d) $t = 13.0$

Figure 19: Continued.

A three-dimensional problem, involving shock interaction with a longitudinal vortex, is studied in Zhang, Zhang and Shu, JFM 2009.



$t = 11$

Figure 20: The streamline pattern and its time evolution in the meridional plane for $M_1 = 2$ and $\epsilon = 7$.

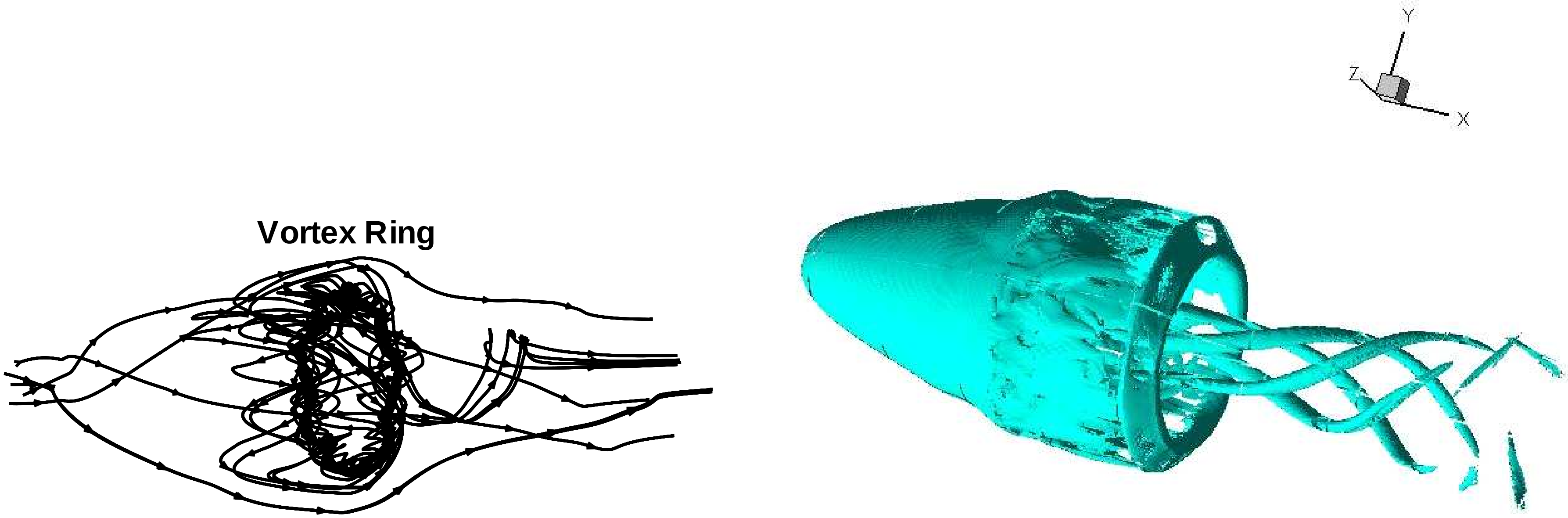


Figure 21: Three dimensional streamlines (left) and iso-vorticity surface $|\omega| = 9$ at $t = 11$.

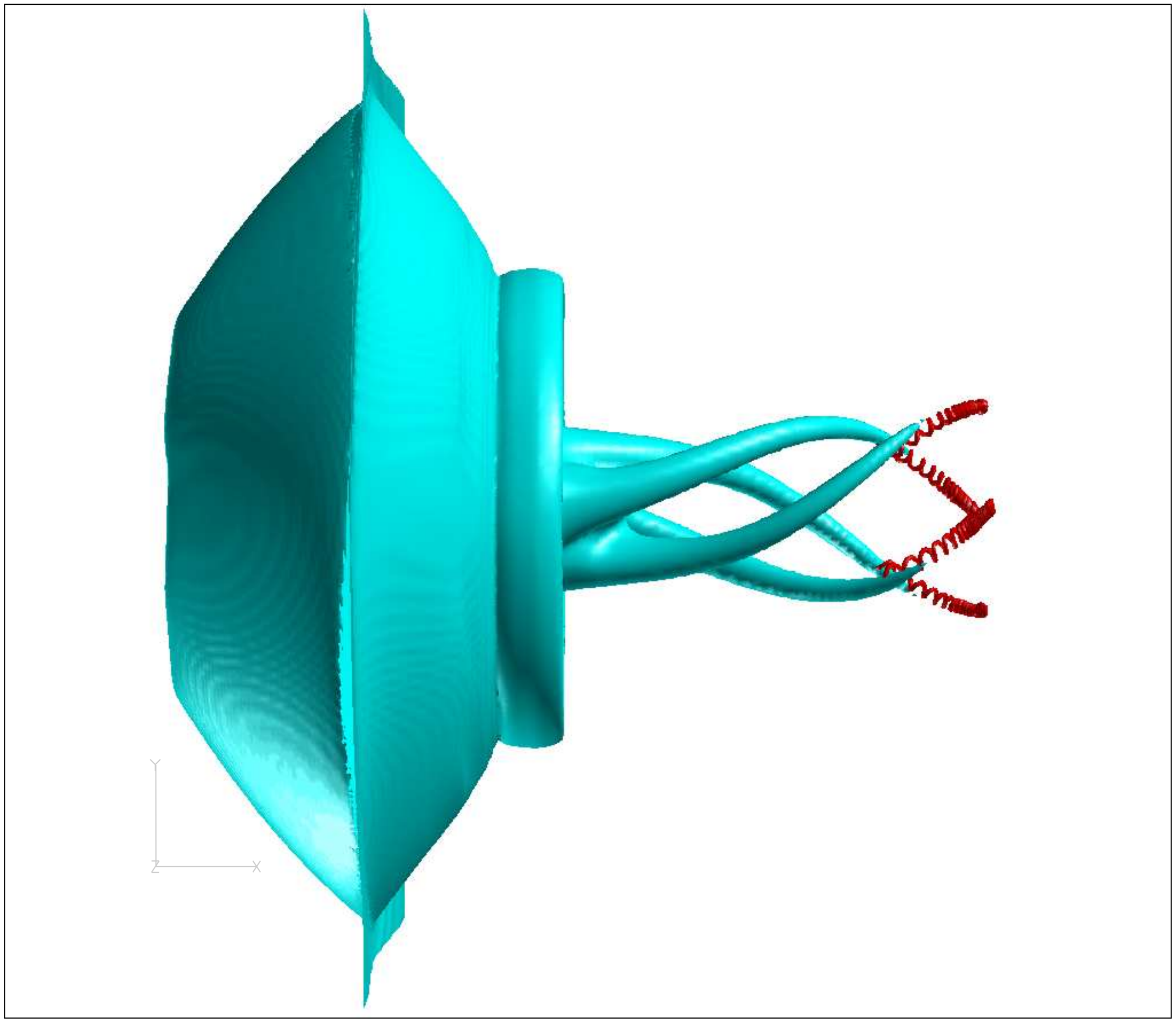


Figure 22: The iso-surface of density $\rho = 1.75$ and three-dimensional streamlines (in red) at $t = 11$.

References:

S. Zhang, Y.-T. Zhang and C.-W. Shu, *Interaction of a shock wave with an oblique vortex pair: shock dynamics and mechanism of sound generation*, Physics of Fluids, v18 (2006), article number 126101.

S. Zhang, S. Jiang, Y.-T. Zhang and C.-W. Shu, *The mechanism of sound generation in the interaction between a shock wave and two counter rotating vortices*, Physics of Fluids, v21 (2009), article number 076101.

S. Zhang, H. Zhang and C.-W. Shu, *Topological structure of shock induced vortex breakdown*, Journal of Fluid Mechanics, v639 (2009), pp.343-372.

S. Zhang, H. Li, X. Liu, H. Zhang and C.-W. Shu, *Classification and sound generation of two-dimensional interaction of two Taylor vortices*, Physics of Fluids, v25 (2013), 056103.