DG method I: hyperbolic conservation laws

Chi-Wang Shu

Division of Applied Mathematics

Brown University

FlowModellium Lab, MIPT, September 2013

Outline

- The first DG scheme in 1973 for neutron transport
- Setup of Runge-Kutta DG schemes
- Properties of DG schemes
- Systems and multi-dimensions, unstructured meshes
- History and references
- Remarks on implementations: matrix form for linear equations, quadratures and quadrature-free, CPR schemes

- Recent development and applications:

  1. A simple WENO limiter for DG schemes
  2. Positivity-preserving DG and finite volume schemes
  3. DG method for problems involving δ-singularities

The first DG scheme for neutron transport

The first discontinuous Galerkin method was introduced in 1973 by Reed and Hill (Los Alamos Technical Report), for neutron transport, i.e. a time independent linear hyperbolic equation.

Consider the two dimensional steady state linear convection equation

$$au_x + bu_y = f(x, y), \qquad 0 \le x,\, y \le 1, \tag{1}$$

where a and b are constants. Without loss of generality we assume $a > 0, b > 0$.

The equation (1) is well-posed when equipped with the inflow boundary condition

$$u(x,0) = g_1(x), \;\; 0 \le x \le 1 \qquad \text{and} \qquad u(0,y) = g_2(y), \;\; 0 \le y \le 1. \tag{2}$$

For simplicity, we assume a rectangular mesh to cover the computational domain $[0,1]^2$, consisting of cells

$$I_{i,j} = \{(x,y) : \; x_{i-\frac{1}{2}} \le x \le x_{i+\frac{1}{2}}, \; y_{j-\frac{1}{2}} \le y \le y_{j+\frac{1}{2}}\}$$

for $1 \le i \le N_x$ and $1 \le j \le N_y$, where

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N_x+\frac{1}{2}} = 1$$

and

$$0 = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \cdots < y_{N_y+\frac{1}{2}} = 1.$$

We denote

$$\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \ \ 1 \le i \le N_x; \qquad \Delta y_j = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \ \ 1 \le j \le N_y;$$

and

$$h = \max \left(\max_{1 \le i \le N_x} \Delta x_i, \ \max_{1 \le j \le N_y} \Delta y_j \right).$$

We assume the mesh is regular, namely there is a constant $c > 0$ independent of h such that

$$\Delta x_i \ge ch, \ \ 1 \le i \le N_x; \qquad \Delta y_j \ge ch, \ \ 1 \le j \le N_y.$$

We define a finite element space consisting of piecewise polynomials

$$V_h^k = \left\{ v : v|_{I_{i,j}} \in P^k(I_{i,j}); \ 1 \le i \le N_x, 1 \le j \le N_y \right\}, \qquad (3)$$

where $P^k(I_{i,j})$ denotes the set of polynomials of degree up to k on $I_{i,j}$.

The discontinuous Galerkin (DG) method for solving (1) is defined as follows: find the unique function $u_h \in V_h^k$ such that, for all test functions $v_h \in V_h^k$ and all $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$, we have

$$\begin{aligned} &- \int\int_{I_{i,j}} \left(a u_h (v_h)_x + b u_h (v_h)_y\right) dxdy \\ &+a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{u}_h(x_{i+\frac{1}{2}}, y) v_h(x^-_{i+\frac{1}{2}}, y) dy \\ &-a \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{u}_h(x_{i-\frac{1}{2}}, y) v_h(x^+_{i-\frac{1}{2}}, y) dy \\ &+b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{u}_h(x, y_{j+\frac{1}{2}}) v_h(x, y^-_{j+\frac{1}{2}}) dx \\ &-b \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{u}_h(x, y_{j-\frac{1}{2}}) v_h(x, y^+_{j-\frac{1}{2}}) dx = \int\int_{I_{i,j}} f\, v_h\, dxdy. \end{aligned} \tag{4}$$

Here, $\hat{u}_h$ is the so-called “numerical flux”, depending on the values of u_h at both sides of the interface.

For the simple linear convection PDE (1), the numerical flux can be chosen according to the upwind principle, namely

$$\hat{u}_h(x_{i+\frac{1}{2}}, y) = u_h(x^-_{i+\frac{1}{2}}, y), \qquad \hat{u}_h(x, y_{j+\frac{1}{2}}) = u_h(x, y^-_{j+\frac{1}{2}}).$$

Notice that, for the boundary cell $i = 1$, the numerical flux for the left edge is defined using the given boundary condition

$$\hat{u}_h(x_{\frac{1}{2}}, y) = g_2(y).$$

Likewise, for the boundary cell $j = 1$, the numerical flux for the bottom edge is defined by

$$\hat{u}_h(x, y_{\frac{1}{2}}) = g_1(x).$$

Now, for $k = 0$ (piecewise constant), the scheme becomes the well-known first order upwind scheme:

$$a\left(\frac{u_{i,j} - u_{i-1,j}}{\Delta x}\right) + b\left(\frac{u_{i,j} - u_{i,j-1}}{\Delta y}\right) = f(x_i, y_j).$$

Clearly, if we know the boundary data $u(i, 0)$ and $u(0, j)$, we can explicitly compute sequentially $u(1, 1)$, $u(2, 1)$, $u(1, 2)$, etc.

In general, If a local basis of $P^k(I_{i,j})$ is chosen and denoted as $\varphi^\ell_{i,j}(x,y)$ for $\ell = 1, 2, \cdots, K = (k+1)(k+2)/2$, we can express the numerical solution as

$$u_h(x,y) = \sum_{\ell=1}^{K} u^\ell_{i,j}\varphi^\ell_{i,j}(x,y), \qquad (x,y) \in I_{i,j},$$

and we should solve for the coefficients

$$u_{i,j} = \begin{pmatrix} u^1_{i,j} \\ \vdots \\ u^K_{i,j} \end{pmatrix},$$

which, according to the scheme (4), satisfies the linear equation

$$A_{i,j}u_{i,j} = rhs \tag{5}$$

where $A_{i,j}$ is a $K \times K$ matrix whose (ℓ, m)-th entry is given by

$$a_{i,j}^{\ell,m} = -\int\int_{I_{i,j}} \Big(a\varphi_{i,j}^{m}(x,y)(\varphi_{i,j}^{\ell}(x,y))_x + b\varphi_{i,j}^{m}(x,y)(\varphi_{i,j}^{\ell}(x,y))_y \Big)\, dxdy + a\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \varphi_{i,j}^{m}(x_{i+\frac{1}{2}},y)\varphi_{i,j}^{\ell}(x_{i+\frac{1}{2}},y)dy + b\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \varphi_{i,j}^{m}(x,y_{j+\frac{1}{2}})\varphi_{i,j}^{\ell}(x,y_{j+\frac{1}{2}})dx, \tag{6}$$

and the ℓ-th entry of the right-hand-side vector is given by

$$\begin{aligned} rhs^{\ell} \;\; = \;\; & a\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} u_h(x_{i-\frac{1}{2}}^{-},y)\varphi_{i,j}^{\ell}(x_{i-\frac{1}{2}},y)dy \\ & +b\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_h(x,y_{j-\frac{1}{2}}^{-})\varphi_{i,j}^{\ell}(x,y_{j-\frac{1}{2}})dx + \int_{I_{i,j}} f\,\varphi_{i,j}^{\ell}\,dxdy, \end{aligned}$$

which depends on the information of u_h in $I_{i-1,j}$ and $I_{i,j-1}$.

It is easy to verify that the matrix $A_{i,j}$ in (5) with entries given by (6) is invertible, hence the numerical solution u_h in the cell $I_{i,j}$ can be easily obtained by solving the small linear system (5), once the solution at the left and bottom cells $I_{i-1,j}$ and $I_{i,j-1}$ are already known, or if one or both of these cells are outside the computational domain.

Therefore, we can obtain the numerical solution u_h in the following ordering: first we obtain it in the cell $I_{1,1}$, since both its left and bottom boundaries are equipped with the prescribed boundary conditions. We then obtain the solution in the cells $I_{2,1}$ and $I_{1,2}$. The next group of cells to be solved are $I_{3,1}$, $I_{2,2}$, $I_{1,3}$. It is clear that we can obtain the solution u_h sequentially in this way for all cells in the computational domain. This method does not involve any large system solvers and is very easy to implement.

The method can be easily used on initial-boundary value problems of linear time dependent hyperbolic equations: we just need to identify the time variable t as one of the spatial variables. It is also easily generalizable to higher dimensions and on arbitrary triangulations. L^2 error estimates of at least $O(h^{k+1/2})$ and sometimes $O(h^{k+1})$ can be proved, Lesaint and Raviart (1974) and Johnson et al (1980s).

It is however difficult to generalize the method to linear systems or to nonlinear problems.

Setup of Runge-Kutta DG schemes

We are interested in solving a hyperbolic conservation law

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

In 2D it is

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

and in system cases u is a vector, and the Jacobian $f'(u)$ is diagonalizable with real eigenvalues.

Several properties of the solutions to hyperbolic conservation laws.

- The solution u may become discontinuous regardless of the smoothness of the initial conditions.
- Weak solutions are not unique. The unique, physically relevant entropy solution satisfies additional entropy inequalities

$$U(u)_t + F(u)_x \leq 0$$

in the distribution sense, where $U(u)$ is a convex scalar function of u and the entropy flux $F(u)$ satisfies $F'(u) = U'(u)f'(u)$.

To solve the hyperbolic conservation law:

$$u_t + f(u)_x = 0, \tag{7}$$

we multiply the equation with a test function v, integrate over a cell $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, and integrate by parts:

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + f(u_{j+\frac{1}{2}}) v_{j+\frac{1}{2}} - f(u_{j-\frac{1}{2}}) v_{j-\frac{1}{2}} = 0$$

Now assume both the solution u and the test function v come from a finite dimensional approximation space V_h, which is usually taken as the space of piecewise polynomials of degree up to k:

$$V_h = \left\{ v : v|_{I_j} \in P^k(I_j), \ j = 1, \cdots, N \right\}$$

However, the boundary terms $f(u_{j+\frac{1}{2}})$, $v_{j+\frac{1}{2}}$ etc. are not well defined when u and v are in this space, as they are discontinuous at the cell interfaces.

From the conservation and stability (upwinding) considerations, we take

- A single valued monotone numerical flux to replace $f(u_{j+\frac{1}{2}})$:

  $$\hat{f}_{j+\frac{1}{2}} = \hat{f}(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}})$$

  where $\hat{f}(u,u) = f(u)$ (consistency); $\hat{f}(\uparrow, \downarrow)$ (monotonicity) and $\hat{f}$ is Lipschitz continuous with respect to both arguments.

- Values from inside I_j for the test function v

  $$v^-_{j+\frac{1}{2}}, \qquad v^+_{j-\frac{1}{2}}$$

Hence the DG scheme is: find $u \in V_h$ such that

$$\int_{I_j} u_t v dx - \int_{I_j} f(u) v_x dx + \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0 \qquad (8)$$

for all $v \in V_h$.

Notice that, for the piecewise constant $k = 0$ case, we recover the well known first order monotone finite volume scheme:

$$(u_j)_t + \frac{1}{h}\left(\hat{f}(u_j, u_{j+1}) - \hat{f}(u_{j-1}, u_j)\right) = 0.$$

Time discretization could be by the TVD Runge-Kutta method (Shu and Osher, JCP 88). For the semi-discrete scheme:

$$\frac{du}{dt} = L(u)$$

where $L(u)$ is a discretization of the spatial operator, the third order TVD Runge-Kutta is simply:

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

Properties of DG schemes

The DG scheme has the following properties:

- Easy handling of complicated geometry and boundary conditions (common to all finite element methods). Allowing hanging nodes in the mesh (unique to DG);
- Compact. Communication only with immediate neighbors, regardless of the order of the scheme;

- Explicit. Because of the discontinuous basis, the mass matrix is local to the cell, resulting in explicit time stepping (no systems to solve);
- Parallel efficiency. Achieves 99% parallel efficiency for static mesh and over 80% parallel efficiency for dynamic load balancing with adaptive meshes (Flaherty et al.); Also friendly to GPU environment (Klockner et al, JCP10).

- Provable cell entropy inequality and L^2 stability, for arbitrary nonlinear equations in any spatial dimension and any triangulation, for any polynomial degrees, without limiters or assumption on solution regularity (Jiang and Shu, Math. Comp. 94 (scalar case); Hou and Liu, JSC 07 (symmetric systems)). For $U(u) = \frac{u^2}{2}$:

$$\frac{d}{dt}\int_{I_j} U(u)dx + \hat{F}_{j+1/2} - \hat{F}_{j-1/2} \leq 0$$

Summing over j: $\frac{d}{dt}\int_a^b u^2 dx \leq 0.$

This also holds for fully discrete RKDG methods with third order TVD Runge-Kutta time discretization, for linear equations (Zhang and Shu, SINUM 10).

- At least $(k+\frac{1}{2})$-th order accurate, and often $(k+1)$-th order accurate for smooth solutions when piecewise polynomials of degree k are used, regardless of the structure of the meshes, for smooth solutions (Lesaint and Raviart 74; Johnson and Pitkäranta, Math. Comp. 86 (linear steady state); Zhang and Shu, SINUM 04 and 06 (RKDG for nonlinear equations)).

- $(2k+1)$-th order superconvergence in negative norm and in strong L^2-norm for post-processed solution for linear and nonlinear equations with smooth solutions (Cockburn, Luskin, Shu and Süli, Math. Comp. 03; Ryan, Shu and Atkins, SISC 05; Curtis, Kirby, Ryan and Shu, SISC 07; Ji, Xu and Ryan, JSC 13).

- $(k+3/2)$-th or $(k+2)$-th order superconvergence of the DG solution to a special projection of the exact solution, and non-growth of the error in time up to $t = O(\frac{1}{\sqrt{h}})$ or $t = O(\frac{1}{h})$, for linear and nonlinear hyperbolic and convection diffusion equations (Cheng and Shu, JCP 08; Computers & Structures 09; SINUM 10; Meng, Shu, Zhang and Wu, SINUM 12 (nonlinear); Yang and Shu, SINUM 12 ($(k+2)$-th order)).

- Several formulations of DG methods for solving nonlinear Hamilton-Jacobi equations

$$\varphi_t + H(\varphi_x, \varphi_y) = 0$$

  - Using the DG method for the system satisfied by $(u, v) = (\varphi_x, \varphi_y)$ (Hu and Shu, SISC 99; Li and Shu, Applied Mathematics Letters 05; Xiong, Shu and Zhang, IJNAM 13).
  - Directly solving for φ (Cheng and Shu, JCP 07; Bokanowski, Cheng and Shu, SISC 11; Num. Math. to appear; Xiong, Shu and Zhang, IJNAM 13).
  - An LDG method (Yan and Osher, JCP 11; Xiong, Shu and Zhang, IJNAM 13).

- Easy h-p adaptivity.
- Stable and convergent DG methods are now available for many nonlinear PDEs containing higher derivatives: convection diffusion equations, KdV equations, ...

Systems and multi-dimensions, unstructured meshes

The RKDG method applies in the same form to hyperbolic systems. The only difference is that monotone numerical fluxes are replaced by numerical fluxes based on exact or approximate Riemann solvers (Godunov, Lax-Friedrichs, HLLC, etc. See the book of Toro). Local characteristic decomposition is not needed unless a nonlinear limiter is used.

The RKDG method applies in the same way to multi-dimensional problems including unstructured meshes. Integration by parts is replaced by divergence theorem. Numerical fluxes are still one-dimensional in the normal direction of the cell boundary.

History and references

Here is a (very incomplete) history of the early study of DG methods for convection dominated problems:

- 1973: First discontinuous Galerkin method for steady state linear scalar conservation laws (Reed and Hill).
- 1974: First error estimate (for tensor product mesh) of the discontinuous Galerkin method of Reed and Hill (Lesaint and Raviart).
- 1986: Error estimates for discontinuous Galerkin method of Reed and Hill (Johnson and Pitkäranta).
- 1989-1998: Runge-Kutta discontinuous Galerkin method for *nonlinear* conservation laws (Cockburn, Shu, ...).

- 1994: Proof of cell entropy inequality for discontinuous Galerkin method for nonlinear conservation laws in general multidimensional triangulations (Jiang and Shu).
- 1997-1998: Discontinuous Galerkin method for convection diffusion problems (Bassi and Rebay, Cockburn and Shu, Baumann and Oden, ...).
- 2002: Discontinuous Galerkin method for partial differential equations with third or higher order spatial derivatives (KdV, biharmonic, ...) (Yan and Shu, Xu and Shu, ...)

Collected works on the DG methods:

- Discontinuous Galerkin Methods: Theory, Computation and Applications, B. Cockburn, G. Karniadakis and C.-W. Shu, editors, Lecture Notes in Computational Science and Engineering, volume 11, Springer, 2000. (Proceedings of the first DG Conference)
- Journal of Scientific Computing, special issue on DG methods, 2005.
- Computer Methods in Applied Mechanics and Engineering, special issue on DG methods, 2006.
- Journal of Scientific Computing, special issue on DG methods, 2009.
- Li, Discontinuous Finite Elements in Fluid Dynamics and Heat Transfer, Birkhauser 2006.

- Kanschat, Discontinuous Galerkin Methods for Viscous Flow, Deutscher Universitätsverlag, Wiesbaden 2007.
- Hesthaven and Warburton, Nodal Discontinuous Galerkin Methods, Springer 2008.
- Rivière, Discontinuous Galerkin methods for solving elliptic and parabolic equations. Theory and implementation, SIAM 2008.
- Shu, Discontinuous Galerkin methods: general approach and stability, in S. Bertoluzza, S. Falletta, G. Russo, and C.-W. Shu, editors, *Numerical Solutions of Partial Differential Equations*, pages 149–201. Birkhäuser 2009.

Remarks on implementations

To implement the DG method, we need to first choose a local basis of $P^k(I_j)$. Let us denote it is $\varphi_j^\ell(x)$ for $\ell = 0, 1, \cdots, k$.

Examples of local basis:

- Modal basis: we can take a locally orthogonal basis, that is

  $$\int_{I_j} \varphi_j^\ell(x)\varphi_j^m(x)dx = \delta_{\ell m}$$

  Notice that, since I_j is a small cell, this is easy to construct even for unstructured meshes, e.g. through a Gram-Schmidt orthogonalization process.

- Nodal basis: we can take $k+1$ points (usually some kind of quadrature points) and then build a nodal basis in which $\varphi_j^\ell(x)$ is one at the ℓ-th point in this set and zero at all other points in this set.
- Taylor basis: we can just take $\varphi_j^\ell(x) = \left(\frac{x-x_j}{h}\right)^\ell$ where x_j is the center of I_j. This basis is the easiest to construct and works fine for modest k (up to 7 or 8).

Note: The choice of basis does not change the DG algorithm, which is already determined by the choice of the finite element space V_h^k and the numerical fluxes. However, it may affect the efficiency and round-off error accumulation of the DG code. The choice of basis is particularly important for higher order DG, e.g. spectral element methods.

After the basis is chosen, the numerical solution can be written as

$$u_h(x,t) = \sum_{\ell=0}^{k} u_j^\ell(t)\varphi_j^\ell(x), \qquad x \in I_j,$$

and we should solve for the coefficients

$$u_j(t) = \begin{pmatrix} u_j^0(t) \\ \vdots \\ u_j^k(t) \end{pmatrix},$$

Substituting into the DG scheme

$$\int_{I_j} (u_h)_t v dx - \int_{I_j} f(u_h) v_x dx + \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0$$

we obtain

$$M_j u_j'(t) - \int_{I_j} f(u_h) v_x dx + \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = 0$$

where M_j is a small $(k+1) \times (k+1)$ mass matrix whose (ℓ, m)-th element is $M_j^{\ell,m} = \int_{I_j} \varphi_j^\ell(x) \varphi_j^m(x) dx$. M_j is diagonal if the basis $\varphi_j^\ell(x)$ are orthogonal. Otherwise, it is also easy to invert since it is a very small matrix.

If $f(u) = Cu$ is linear, then the DG scheme can be written as

$$u'_j(t) = A^1_j u_{j-1}(t) + A^2_j u_j(t) + A^3_j u_{j+1}(t)$$

where the small $(k+1) \times (k+1)$ matrices A^m_j can be precomputed. This is then very similar to a finite difference scheme.

For nonlinear $f(u)$, the integral $\int_{I_j} f(u_h) v_x dx$ (or its integrated-by-part version $\int_{I_j} f(u_h)_x v dx$) can be approximated by a numerical quadrature, or in a quadrature-free fashion (Atkins and Shu, AIAA J 98). Many variants of DG schemes, for example the recent CPR schemes (Z.J. Wang, H.T. Huynh), can be considered as DG schemes with the integral approximated by certain quadratures.

Three examples

We show three examples to demonstrate the excellent performance of the DG method.

The first example is the linear convection equation

$$u_t + u_x = 0, \qquad or \qquad u_t + u_x + u_y = 0,$$

on the domain $(0, 2\pi) \times (0, T)$ or $(0, 2\pi)^2 \times (0, T)$ with the characteristic function of the interval $(\frac{\pi}{2}, \frac{3\pi}{2})$ or the square $(\frac{\pi}{2}, \frac{3\pi}{2})^2$ as initial condition and periodic boundary conditions.

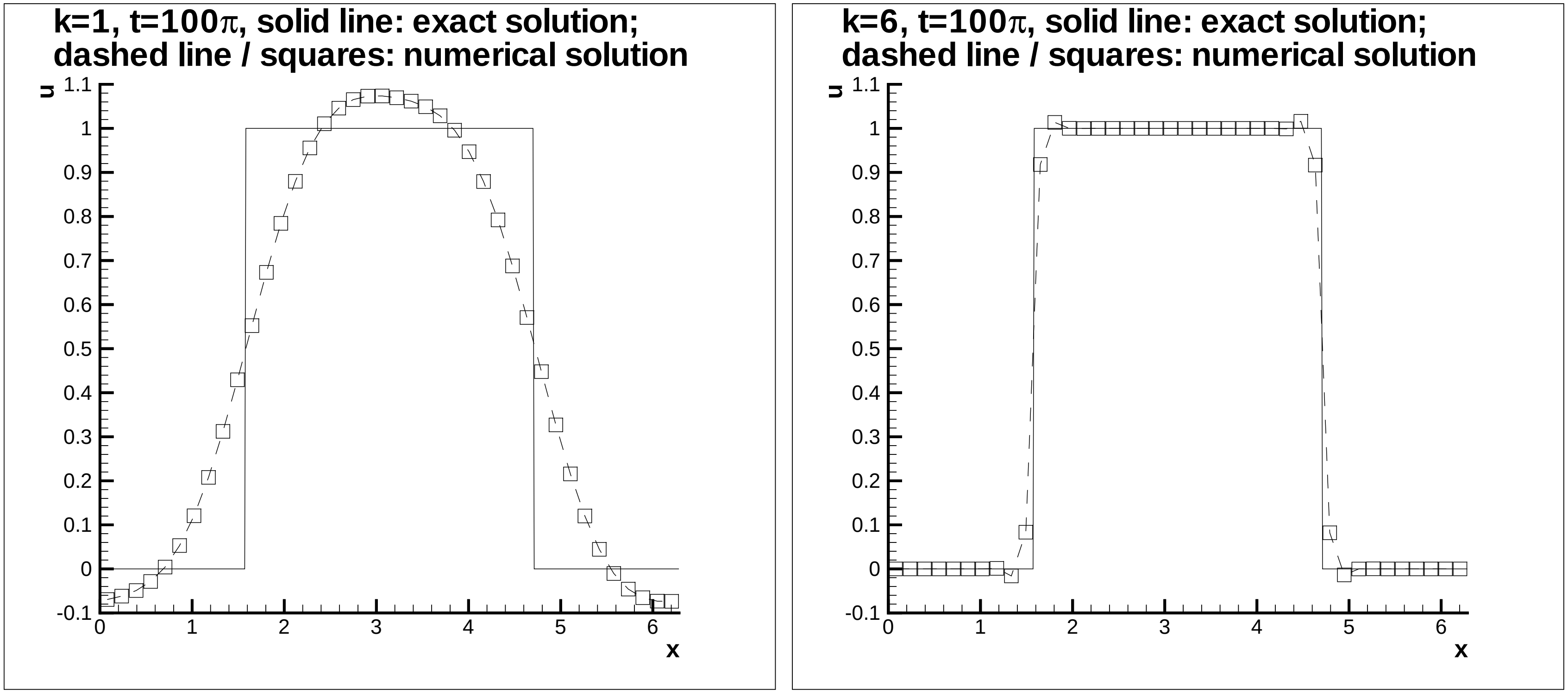


Figure 1: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. One dimensional results with 40 cells, exact solution (solid line) and numerical solution (dashed line and symbols, one point per cell)

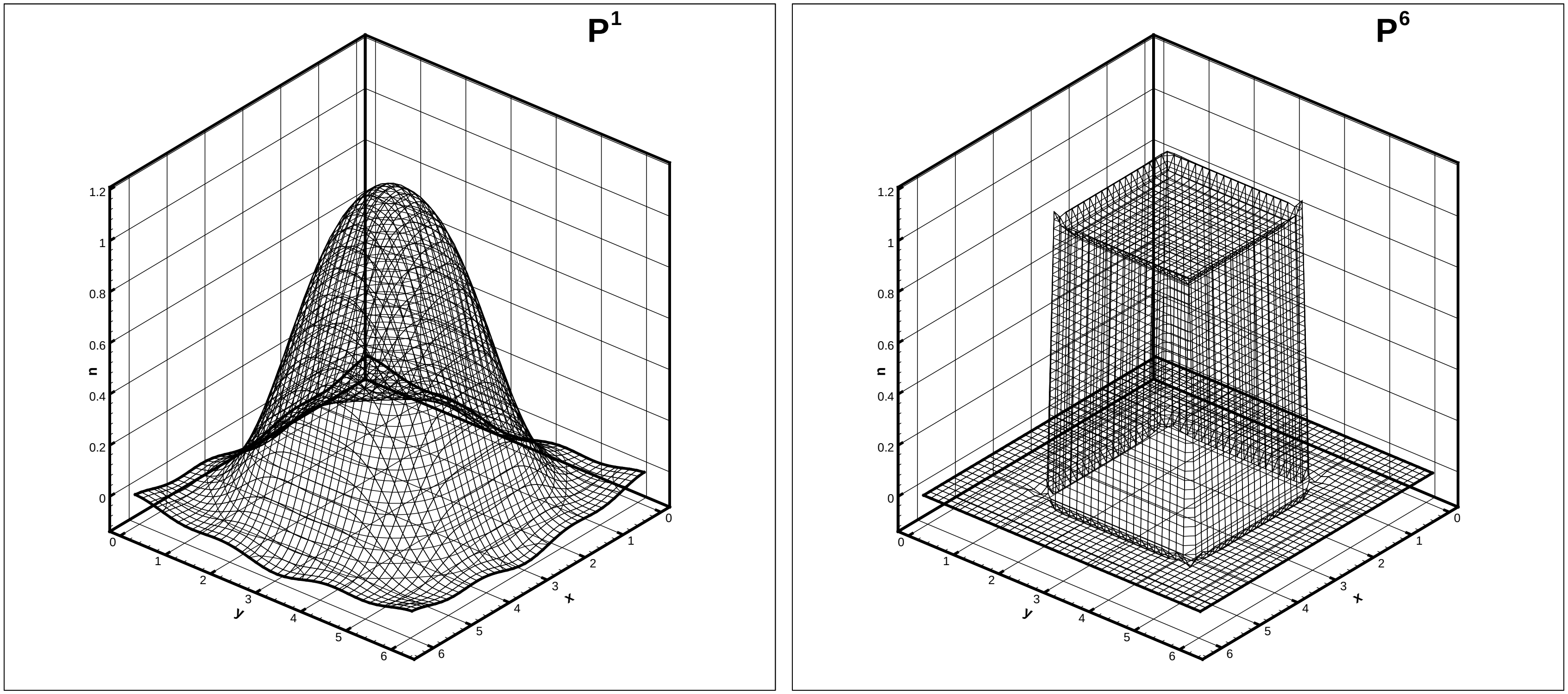


Figure 2: Transport equation: Comparison of the exact and the RKDG solutions at $T = 100\pi$ with second order (P^1, left) and seventh order (P^6, right) RKDG methods. Two dimensional results with 40×40 cells.

The second example is the double Mach reflection problem for the two dimensional compressible Euler equations.

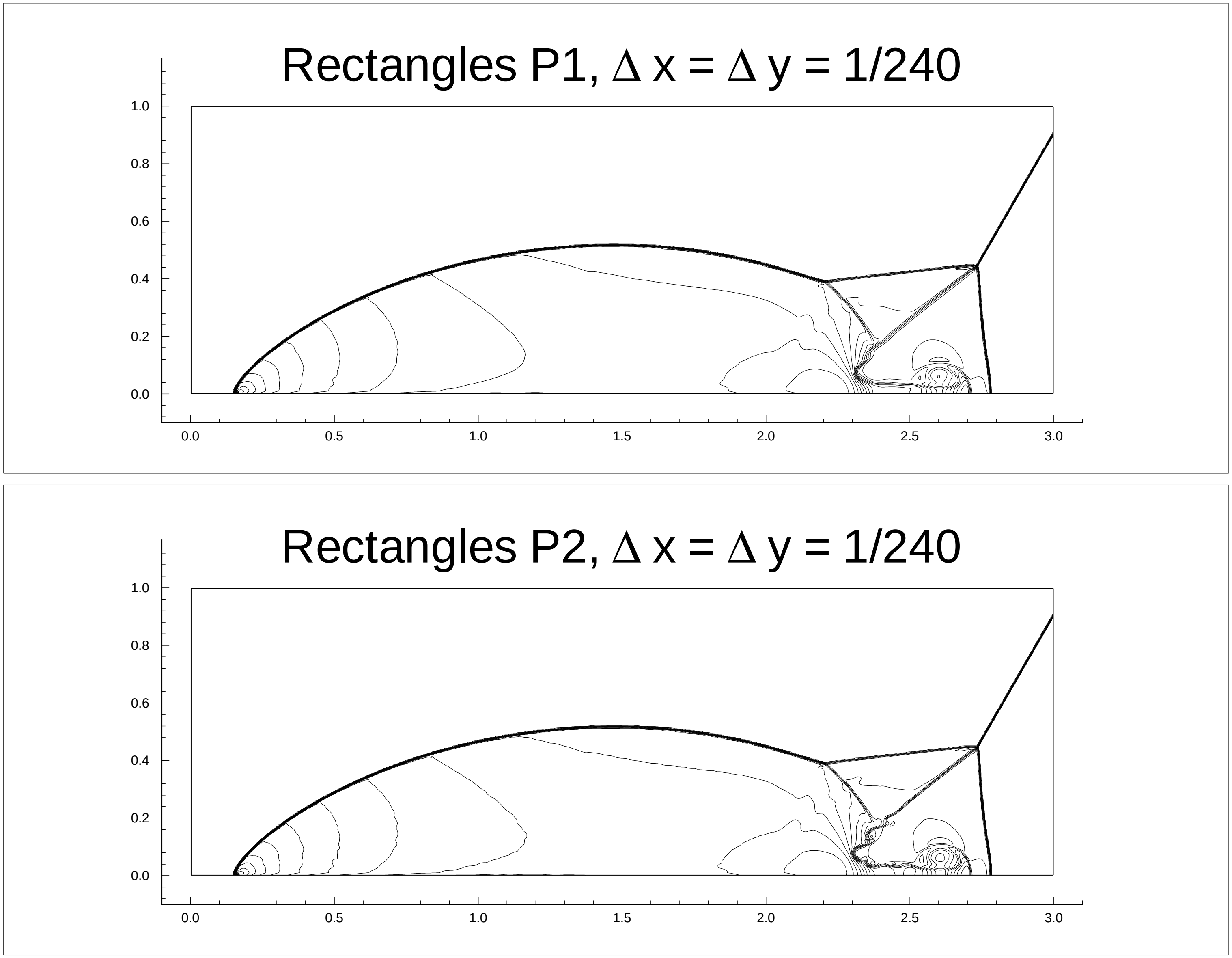


Figure 3: Double Mach reflection. $\Delta x = \Delta y = \frac{1}{240}$. Top: P^1; bottom: P^2.

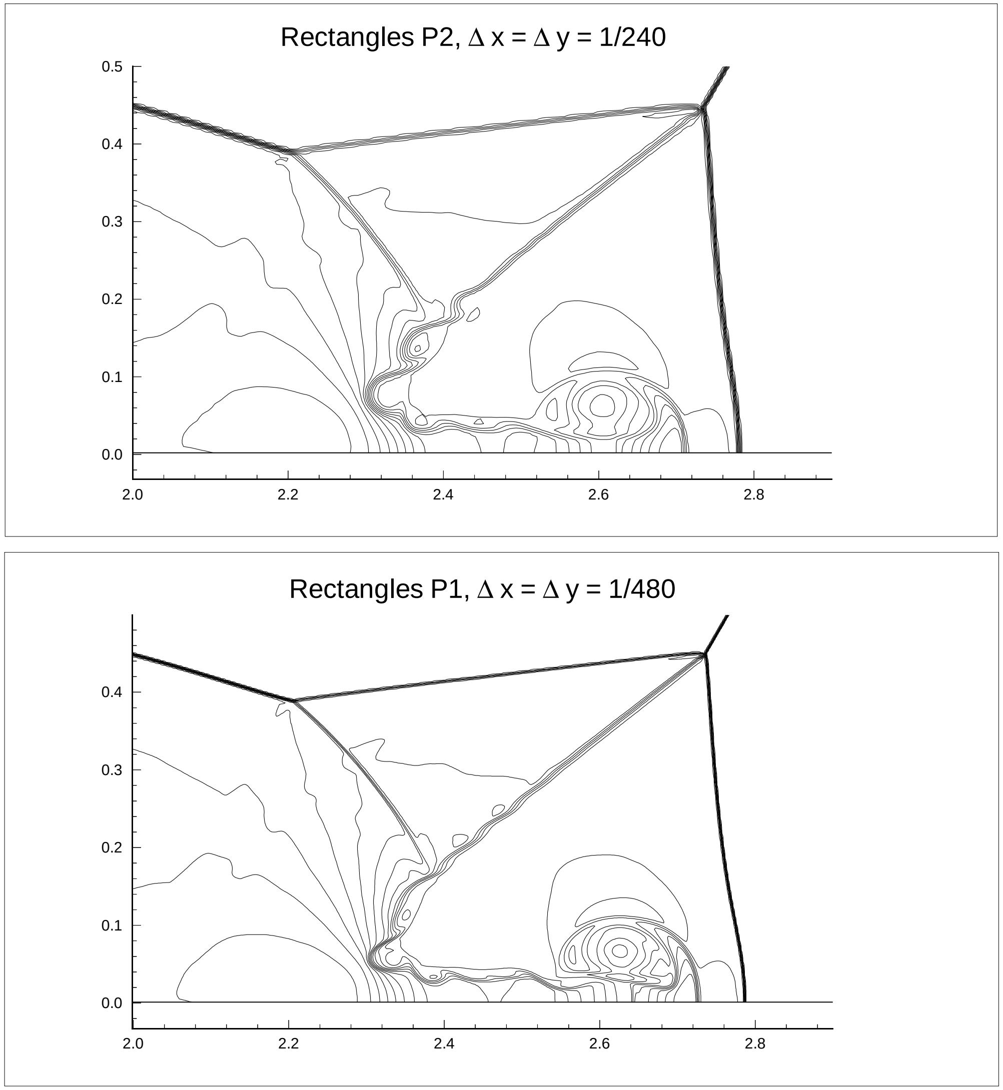


Figure 4: Double Mach reflection. Zoomed-in region. Top: P^2 with $\Delta x = \Delta y = \frac{1}{240}$; bottom: P^1 with $\Delta x = \Delta y = \frac{1}{480}$.

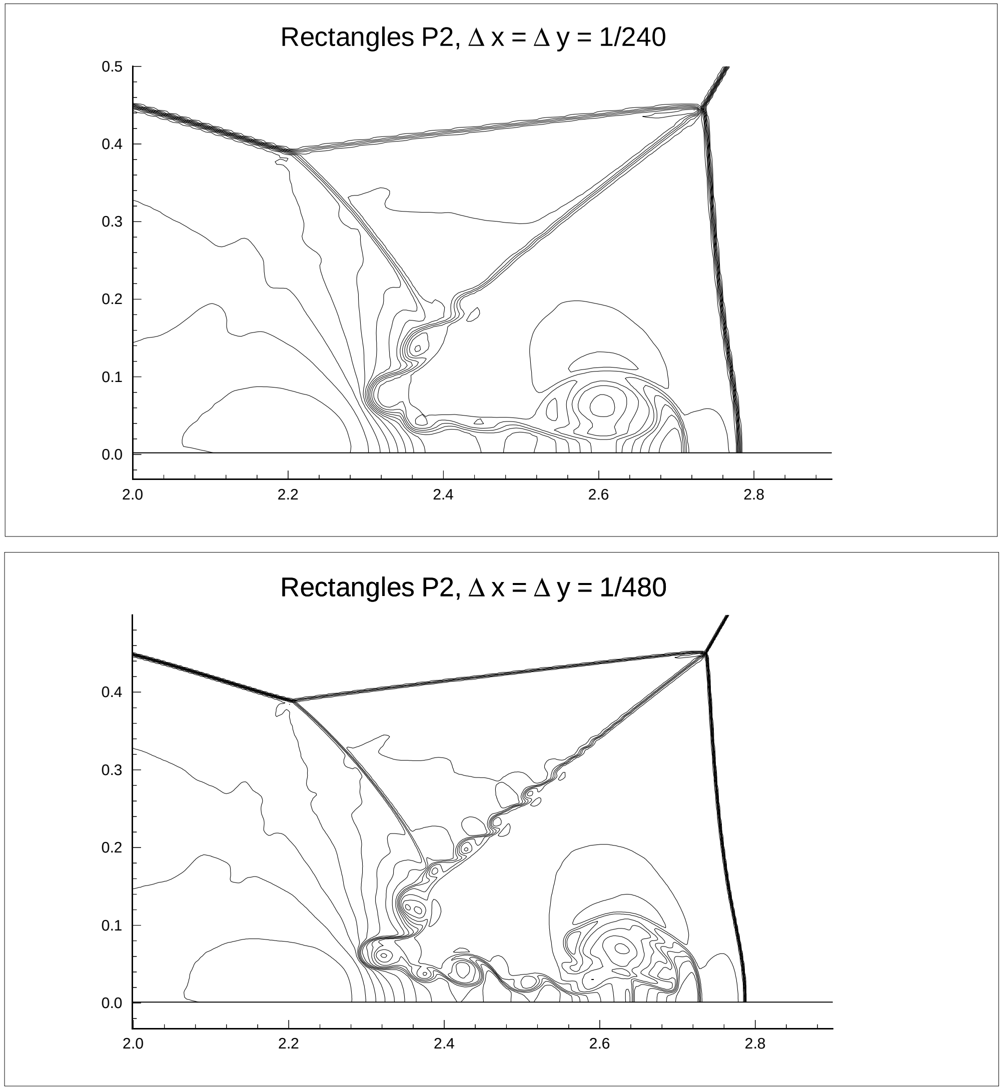
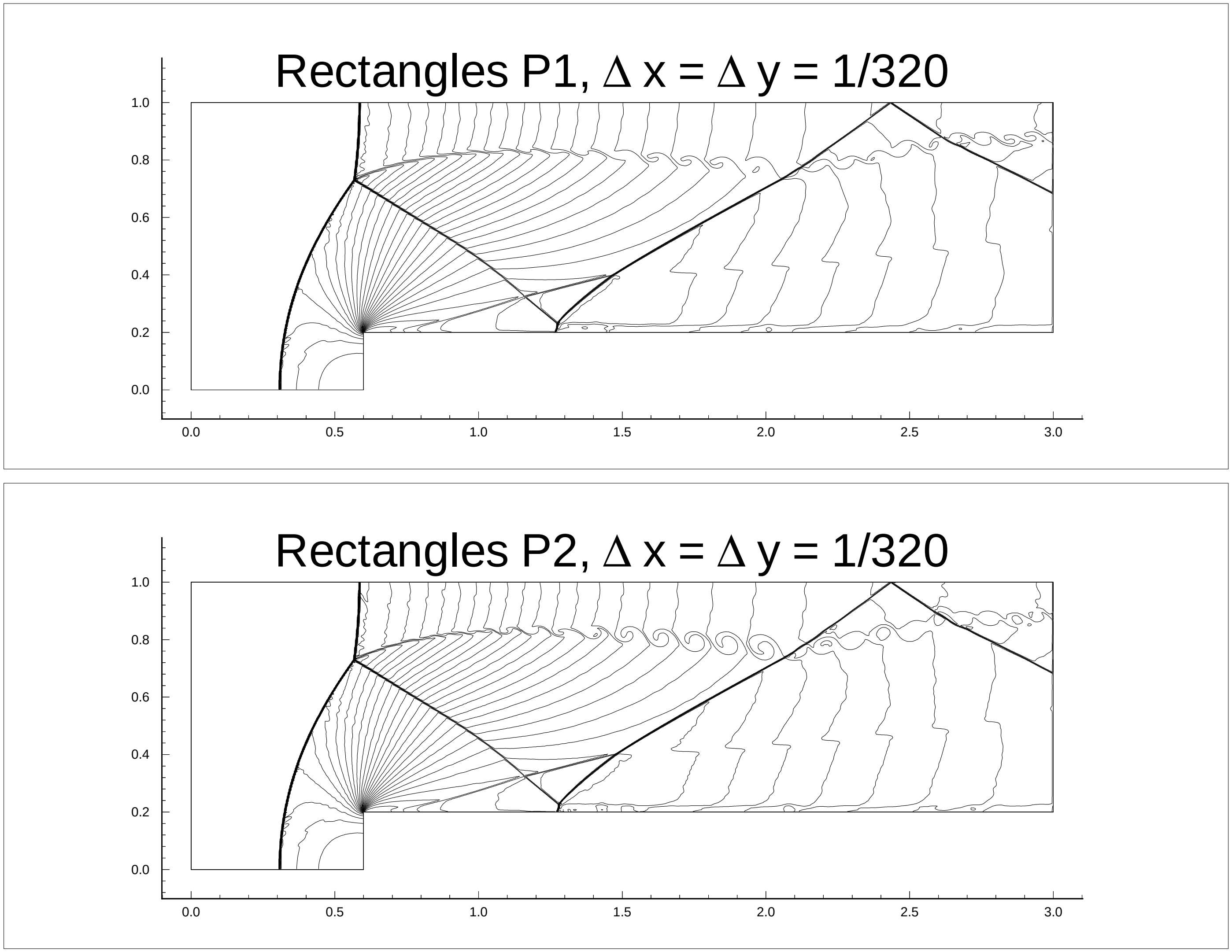


Figure 5: Double Mach reflection. Zoomed-in region. P^2 elements. Top: $\Delta x = \Delta y = \frac{1}{240}$; bottom: $\Delta x = \Delta y = \frac{1}{480}$.

The third example is the flow past a forward-facing step problem for the two dimensional compressible Euler equations. No special treatment is performed near the corner singularity.



Rectangles P1, Δ x = Δ y = 1/320

Rectangles P2, Δ x = Δ y = 1/320

Figure 6: Forward facing step. Zoomed-in region. $\Delta x = \Delta y = \frac{1}{320}$. Left: P^1 elements; right: P^2 elements.

Recent development and applications

A simple WENO limiter for DG schemes

The RKDG schemes for conservation laws are energy stable (L^2 stable). However, for solving problems with strong discontinuities, the DG solution may generate spurious numerical oscillations. In practice, especially for nonlinear problems containing strong shocks, we often need to apply nonlinear limiters to control these oscillations. Most of the limiters studied in the literature come from the methodologies of finite volume high resolution schemes.

A limiter can be considered as a post-processor of the computed DG solution. In any cell which is deemed to contain a possible discontinuity (the so-called troubled cells), the DG polynomial is replaced by a new polynomial of the same degree, while maintaining the original cell average for conservation.

Desired properties of limiters:

- The new polynomial is less oscillatory than the old one
- If the solution in this cell happens to be smooth, then the new polynomial should have the same high order accuracy as the old one.
- The limiter is easy to implement, without parameters which must be adjusted by the user.
- Some provable mathematical properties for the limited solution. For example, TVDM (total variation diminishing in the means); TVB (total variation bounded).

Some commonly used limiters:

- The total variation diminishing (TVD) limiters (Harten, JCP 83).
  - The new polynomial is less oscillatory than the old one.
  - If the solution in this cell happens to be smooth but is near an extrema, then the new polynomial may degenerate to first order accuracy.
  - The limiter is reasonably easy to implement on structured meshes but more difficult to implement on unstructured meshes. It does not involve any user-tuned parameters.
  - The limited scheme is TVDM (total variation diminishing in the means)

- The total variation bounded (TVB) limiters (Shu, Math Comp 87).
  - The new polynomial is less oscillatory than the old one.
  - If the solution in this cell happens to be smooth, then the limiter does not take effect and hence the new polynomial is the same as the old polynomial with of course the same high order accuracy.
  - The limiter is reasonably easy to implement on structured meshes but more difficult to implement on unstructured meshes. It does involve a user-tuned parameter M which is related to the second derivative of the solution at smooth extrema.
  - The limited scheme is TVB (total variation bounded).

- The moment-based limiters (Biswas et al. Appl. Num. Math. 94; Burbeau et al. JCP 01)
  - The new polynomial is less oscillatory than the old one.
  - If the solution in this cell happens to be smooth, the limiter could still degenerate the accuracy to first order.
  - The limiter is reasonably easy to implement on structured meshes but more difficult to implement on unstructured meshes. It does not involve any user-tuned parameters.
  - The limited scheme cannot be proved to be total variation stable.

Some limiters are used in all cells (while they may take effect only in some cells). Examples include the TVD and TVB limiters and the moment-based limiters. Other limiters, in particular the WENO limiters to discussed below, are used only in the so-called troubled cells.

Qiu and Shu (SISC 05) studied several commonly used troubled-cell indicators, and concluded that the TVB indicator (Shu, Math Comp 87) and the KXRCF indicator (Krivodonova et al Appl Num Math 04) are good choices.

There has been a lot of work in the literature on designing WENO limiters for DG schemes. The idea is to have limiters which, even when used erroneously in smooth cells, will not destroy the high order of accuracy. Qiu and Shu, JCP 2003; SISC 2005; Computers & Fluids 2005; Zhu, Qiu, Shu and Dumbser, JCP 2008; Zhu and Qiu, JCP 2012.

The procedure of these WENO limiters is as follows:

- Use a troubled-cell indicator to identify troubled cells. Qiu and Shu, SISC 2005.
- If the cell I_j is identified as a troubled cell, then the DG solution polynomial $p_j(x)$ is replaced by a new polynomial with the same cell average.
  - We first use neighboring cell averages to reconstruct point values at Gaussian points in a regular WENO fashion, then we reconstruct the new polynomial by using the new Gaussian point values and quadrature to compute the moments.
  - The WENO reconstruction could use both cell averages and slopes of neighboring cells (Hermite WENO) to have a narrower stencil.

Drawbacks of earlier WENO limiters:

- Complicated to construct on unstructured meshes (just like regular finite volume WENO schemes on unstructured meshes).
- Information beyond immediate neighbors are needed, especially for high order DG, hence destroying the good local data structure of DG.

The very recent work in Zhong and Shu, JCP 13; Zhu, Zhong, Shu and Qiu, JCP 13 contains a very simple and effective WENO limiter which overcomes both drawbacks mentioned above.

- If the cell I_j is identified as a troubled cell, then the DG solution polynomial $p_j(x)$ is replaced by a convex combination of $p_j(x)$ with $p_{j-1}(x)$ and $p_{j+1}(x)$, the DG solution polynomials of the two immediate neighboring cells. Suitable adjustment is made to ensure that the new polynomials maintains the original cell average (conservation).

- Details:

$$p_j^{new} = w_1 \tilde{p}_{j-1}(x) + w_2 p_j(x) + w_3 \tilde{p}_{j+1}(x)$$

where

$$w_\ell = \frac{\tilde{w}_\ell}{\tilde{w}_1 + \tilde{w}_2 + \tilde{w}_3}; \qquad \tilde{w}_\ell = \frac{\gamma_\ell}{(s_\ell + \varepsilon)^2}$$

with the linear weights given by

$$\gamma_1 = \gamma_3 = \frac{1}{1000}, \quad \gamma_2 = \frac{998}{1000}$$

and the s_ℓ are the standard smoothness indicators of WENO approximations.

Example 1: A Mach 3 wind tunnel with a step. The wind tunnel is 1 length unit wide and 3 length units long. The step is 0.2 length units high and is located 0.6 length units from the left-hand end of the tunnel. The problem is initialized by a right-going Mach 3 flow. Reflective boundary conditions are applied along the wall of the tunnel and inflow/outflow boundary conditions are applied at the entrance/exit.

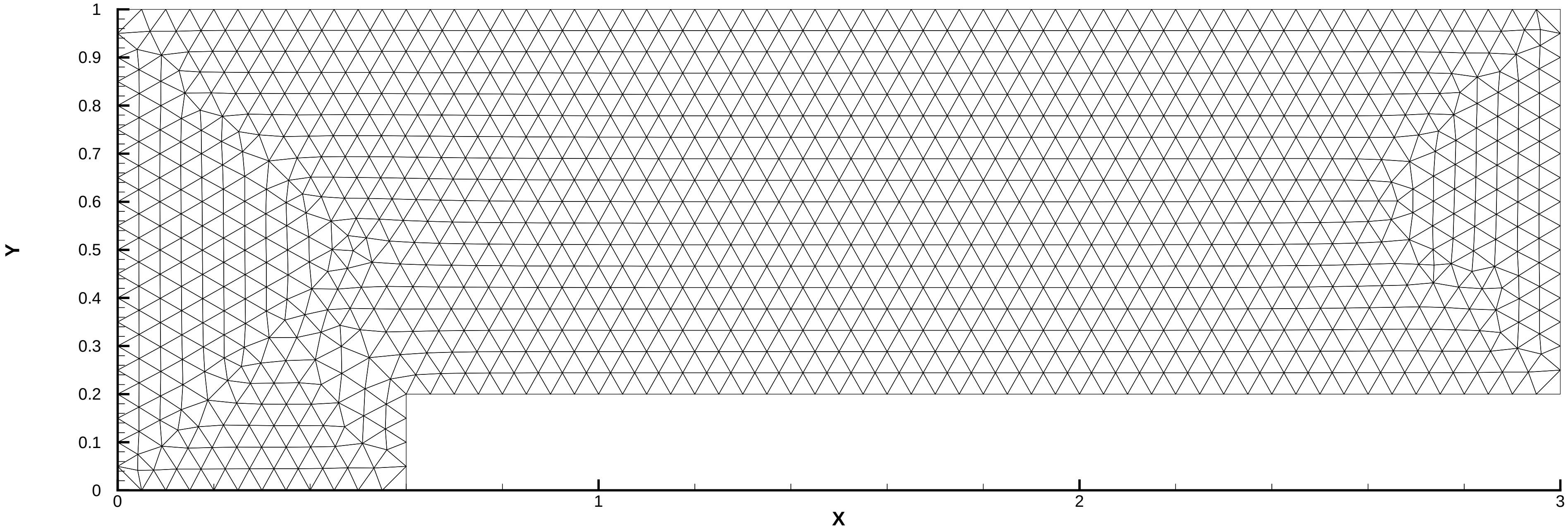


Figure 7: Forward step problem. Sample mesh. The mesh points on the boundary are uniformly distributed with cell length $h = 1/20$.

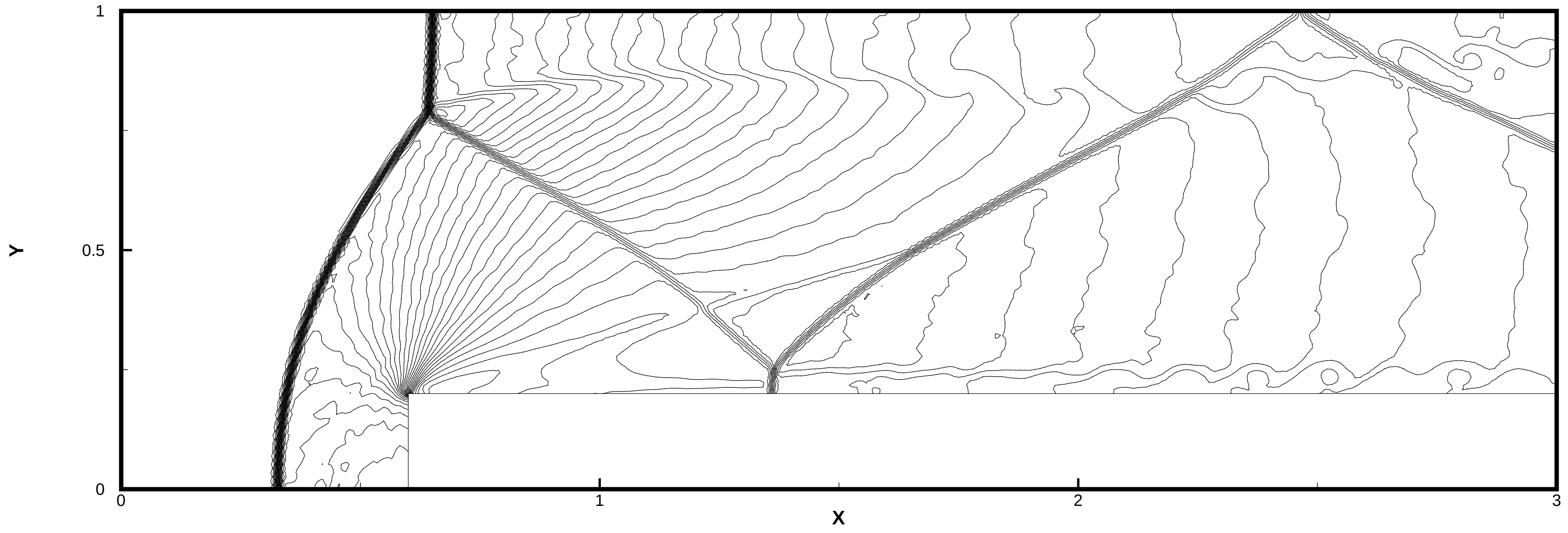


Figure 8: Forward step problem. Third order ($k = 2$) RKDG with the WENO limiter. 30 equally spaced density contours from 0.32 to 6.15. The mesh points on the boundary are uniformly distributed with cell length $h = 1/100$.

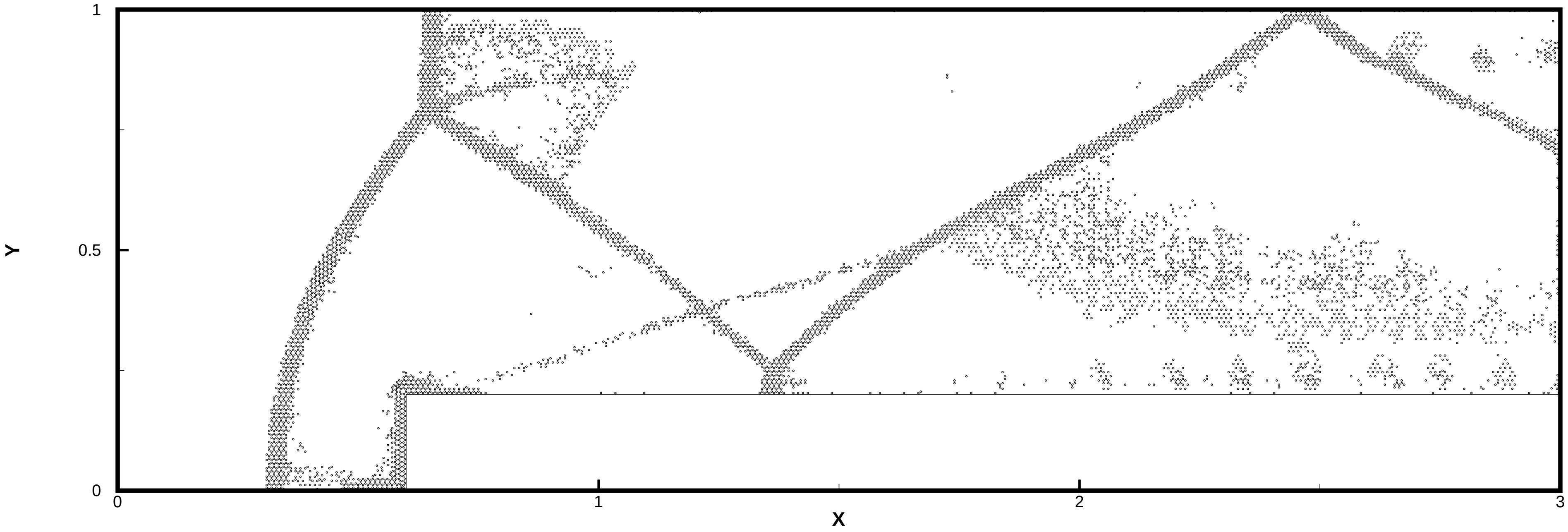


Figure 9: Forward step problem. Third order ($k = 2$) RKDG with the WENO limiter. Troubled cells. Circles denote triangles which are identified as troubled cell subject to the WENO limiting. The mesh points on the boundary are uniformly distributed with cell length $h = 1/100$.

Example 2: We consider inviscid Euler transonic flow past a single NACA0012 airfoil configuration with Mach number $M_\infty = 0.85$, angle of attack $\alpha = 1^\circ$. The computational domain is $[-15, 15] \times [-15, 15]$.

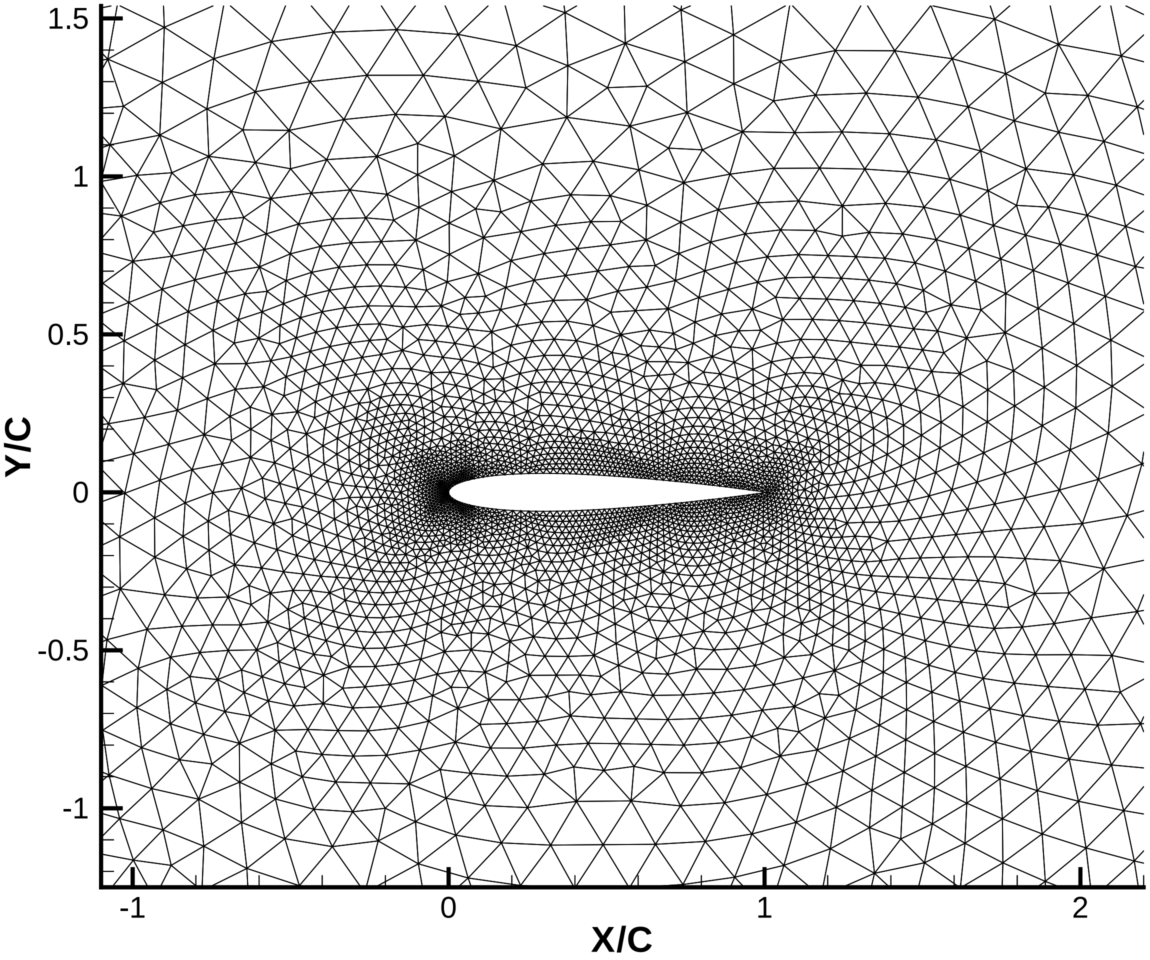


Figure 10: NACA0012 airfoil mesh zoom in.

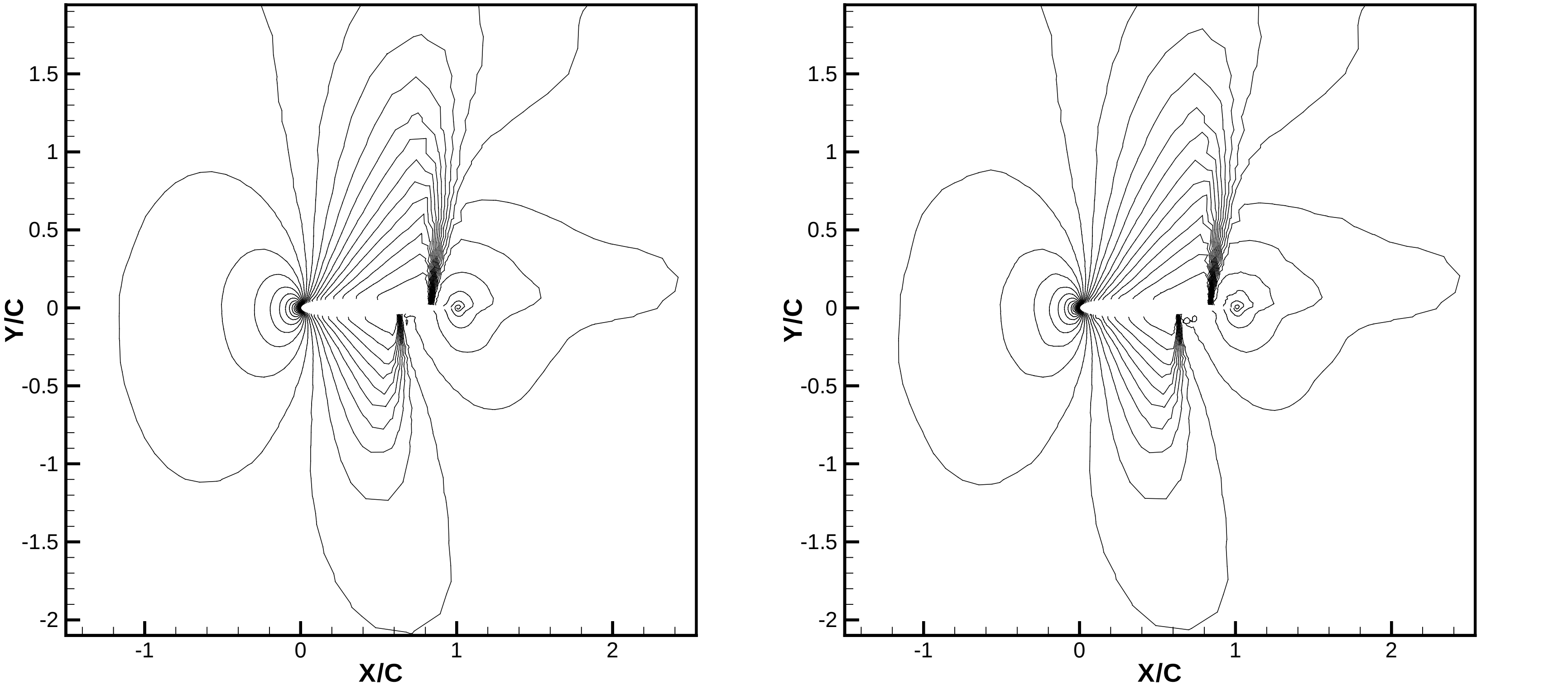


Figure 11: NACA0012 airfoil. Mach number. $M_\infty = 0.85$, angle of attack $\alpha = 1^\circ$, 30 equally spaced mach number contours from 0.158 to 1.357. Left: second order ($k = 1$); right: third order ($k = 2$) RKDG with the WENO limiter.

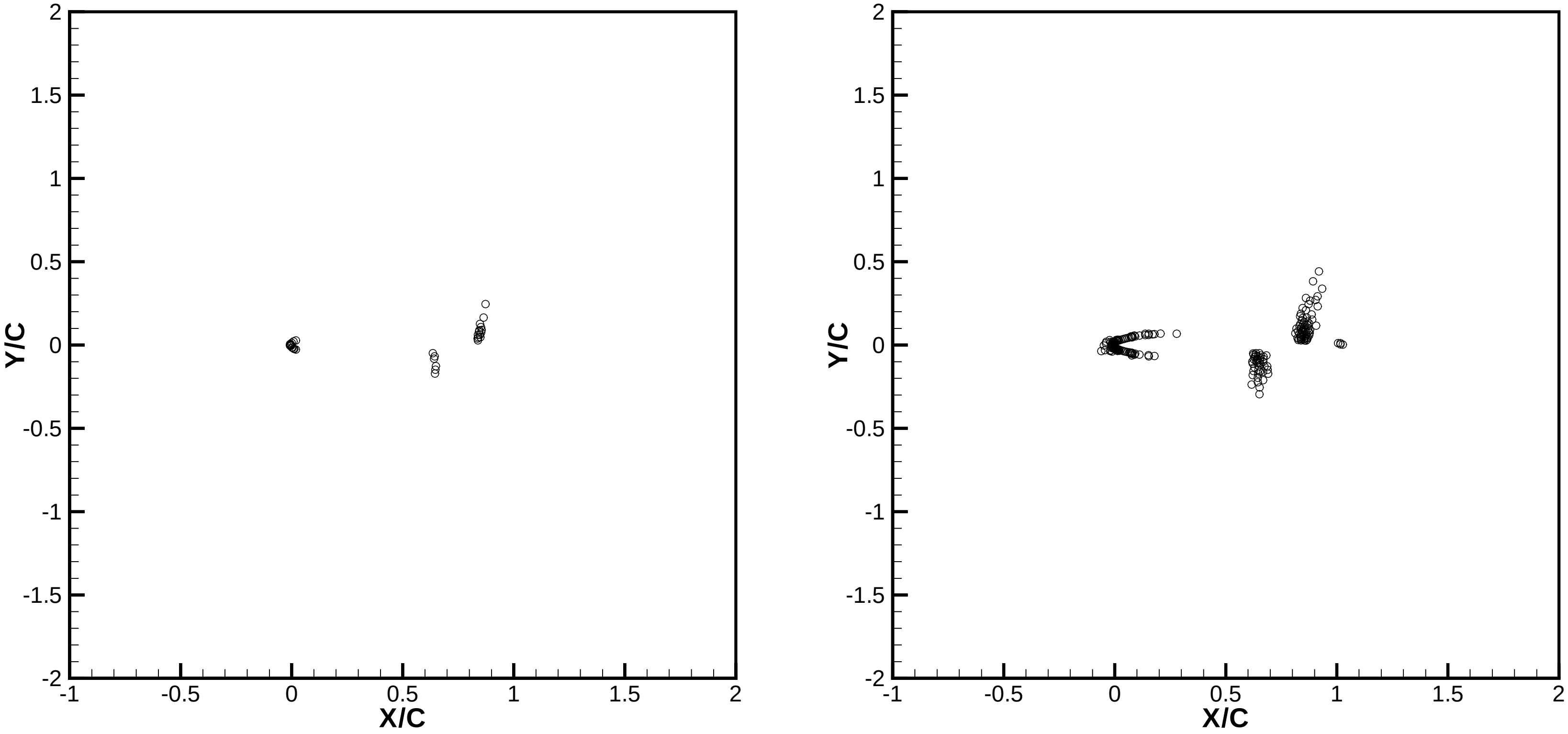


Figure 12: NACA0012 airfoil. Troubled cells. Circles denote triangles which are identified as troubled cells subject to the WENO limiting. $M_\infty = 0.85$, angle of attack $\alpha = 1^\circ$. Left: second order ($k = 1$); right: third order ($k = 2$) RKDG with the WENO limiter.

References:

X. Zhong and C.-W. Shu, *A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods*, Journal of Computational Physics, v232 (2013), pp.397-415.

J. Zhu, X. Zhong, C.-W. Shu and J.-X. Qiu, *Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes*, Journal of Computational Physics, v248 (2013), pp.200-220.

J. Du, C.-W. Shu and M. Zhang, *A simple weighted essentially non-oscillatory limiter for the correction procedure via reconstruction (CPR) framework*, submitted to Applied Numerical Mathematics.

Positivity-preserving DG and finite volume schemes

Introduction

For the scalar conservation laws

$$u_t + \nabla \cdot \mathbf{F}(u) = 0, \qquad u(\mathbf{x}, 0) = u_0(\mathbf{x}). \tag{9}$$

An important property of the entropy solution (which may be discontinuous) is that it satisfies a strict maximum principle: If

$$M = \max_{\mathbf{x}} u_0(\mathbf{x}), \quad m = \min_{\mathbf{x}} u_0(\mathbf{x}), \tag{10}$$

then $u(\mathbf{x}, t) \in [m, M]$ for any $\mathbf{x}$ and t.

First order monotone schemes can maintain the maximum principle. For the one-dimensional conservation law

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

the first order monotone scheme

$$\begin{aligned} u_j^{n+1} &= H_\lambda(u_{j-1}^n, u_j^n, u_{j+1}^n) \\ &= u_j^n - \lambda[h(u_j^n, u_{j+1}^n) - h(u_{j-1}^n, u_j^n)] \end{aligned}$$

where $\lambda = \frac{\Delta t}{\Delta x}$ and $h(u^-, u^+)$ is a monotone flux ($h(\uparrow, \downarrow)$), satisfies

$$H_\lambda(\uparrow, \uparrow, \uparrow)$$

under a suitable CFL condition

$$\lambda \leq \lambda_0.$$

Therefore, if

$$m \leq u_{j-1}^n, u_j^n, u_{j+1}^n \leq M$$

then

$$u_j^{n+1} = H_\lambda(u_{j-1}^n, u_j^n, u_{j+1}^n) \geq H_\lambda(m, m, m) = m,$$

and

$$u_j^{n+1} = H_\lambda(u_{j-1}^n, u_j^n, u_{j+1}^n) \leq H_\lambda(M, M, M) = M.$$

However, for higher order linear schemes, i.e. schemes which are linear for a linear PDE

$$u_t + au_x = 0 \tag{11}$$

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 maximum principle is not satisfied. In fact, no linear schemes with order of accuracy higher than one can satisfy the maximum principle (Godunov Theorem).

Therefore, nonlinear schemes, namely schemes which are nonlinear even for linear PDEs, have been designed to overcome this difficulty. These include roughly two classes of schemes:

- TVD schemes. Most TVD (total variation diminishing) schemes also satisfy strict maximum principle, even in multi-dimensions. TVD schemes can be designed for any formal order of accuracy for solutions in smooth, monotone regions. However, all TVD schemes will degenerate to first order accuracy at smooth extrema.
- TVB schemes, ENO schemes, WENO schemes. These schemes do not insist on strict TVD properties, therefore they do not satisfy strict maximum principles, although they can be designed to be arbitrarily high order accurate for smooth solutions.

Remark: If we insist on the maximum principle interpreted as

$$m \leq u_j^{n+1} \leq M, \qquad \forall j$$

if

$$m \leq u_j^n \leq M, \qquad \forall j,$$

where u_j^n is either the approximation to the point value $u(x_j, t^n)$ for a finite difference scheme, or to the cell average $\frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x, t^n) dx$ for a finite volume or DG scheme, then the scheme can be at most second order accurate (proof due to Harten, see Zhang and Shu, Proceedings of the Royal Society A, 2011).

Therefore, the correct procedure to follow in designing high order schemes that satisfy a strict maximum principle is to change the definition of maximum principle. Note that a high order finite volume scheme has the following algorithm flowchart:

(1) Given $\{\bar{u}_j^n\}$

(2) reconstruct $u^n(x)$ (piecewise polynomial with cell average $\bar{u}_j^n$)

(3) evolve by, e.g. Runge-Kutta time discretization to get $\{\bar{u}_j^{n+1}\}$

(4) return to (1)

Therefore, instead of requiring

$$m \le \bar{u}_j^{n+1} \le M, \qquad \forall j$$

if

$$m \le \bar{u}_j^{n} \le M, \qquad \forall j,$$

we will require

$$m \le u^{n+1}(x) \le M, \qquad \forall x$$

if

$$m \le u^{n}(x) \le M, \qquad \forall x.$$

Similar definition and procedure can be used for discontinuous Galerkin schemes.

Maximum-principle-preserving for scalar conservation laws

The flowchart for designing a high order scheme which obeys a strict maximum principle is as follows:

1. Start with $u^n(x)$ which is high order accurate

$$|u(x,t^n) - u^n(x)| \leq C\Delta x^p$$

and satisfy

$$m \leq u^n(x) \leq M, \qquad \forall x$$

therefore of course we also have

$$m \leq \bar{u}_j^n \leq M, \qquad \forall j.$$

2. Evolve for one time step to get

$$m \le \bar{u}_j^{n+1} \le M, \qquad \forall j. \tag{12}$$

3. Given (12) above, obtain the reconstruction $u^{n+1}(x)$ which

- satisfies the maximum principle

$$m \le u^{n+1}(x) \le M, \qquad \forall x;$$

- is high order accurate

$$|u(x, t^{n+1}) - u^{n+1}(x)| \le C\Delta x^p.$$

Three major difficulties

1. The first difficulty is how to evolve in time for one time step to guarantee

$$m \le \bar{u}_j^{n+1} \le M, \qquad \forall j. \tag{13}$$

   This is very difficult to achieve. Previous works use one of the following two approaches:

- Use exact time evolution. This can guarantee

  $$m \leq \bar{u}_j^{n+1} \leq M, \qquad \forall j.$$

  However, it can only be implemented with reasonable cost for linear PDEs, or for nonlinear PDEs in one dimension. This approach was used in, e.g., Jiang and Tadmor, SISC 1998; Liu and Osher, SINUM 1996; Sanders, Math Comp 1988; Qiu and Shu, SINUM 2008; Zhang and Shu, SINUM 2010; to obtain TVD schemes or maximum-principle-preserving schemes for linear and nonlinear PDEs in one dimension or for linear PDEs in multi-dimensions, for second or third order accurate schemes.

- Use simple time evolution such as SSP Runge-Kutta or multi-step methods. However, additional limiting will be needed on $u^n(x)$ which will destroy accuracy near smooth extrema.

We have figured out a way to obtain

$$m \le \bar{u}_j^{n+1} \le M, \qquad \forall j$$

with simple Euler forward or SSP Runge-Kutta or multi-step methods without losing accuracy on the limited $u^n(x)$:

The evolution of the cell average for a higher order finite volume or DG scheme satisfies

$$\begin{aligned}\bar{u}_j^{n+1} &= G(\bar{u}_j^n, u^-_{j-\frac{1}{2}}, u^+_{j-\frac{1}{2}}, u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) \\ &= \bar{u}_j^n - \lambda[h(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) - h(u^-_{j-\frac{1}{2}}, u^+_{j-\frac{1}{2}})],\end{aligned}$$

where

$$G(\uparrow, \uparrow, \downarrow, \downarrow, \uparrow)$$

therefore there is no maximum principle. The problem is with the two arguments $u^+_{j-\frac{1}{2}}$ and $u^-_{j+\frac{1}{2}}$ which are values at points inside the cell I_j.

The polynomial $p_j(x)$ (either reconstructed in a finite volume method or evolved in a DG method) is of degree k, defined on I_j such that $\bar{u}_j^n$ is its cell average on I_j, $u^+_{j-\frac{1}{2}} = p_j(x_{j-\frac{1}{2}})$ and $u^-_{j+\frac{1}{2}} = p_j(x_{j+\frac{1}{2}})$.

We take a Legendre Gauss-Lobatto quadrature rule which is exact for polynomials of degree k, then

$$\bar{u}_j^n = \sum_{\ell=0}^{m} \omega_\ell p_j(y_\ell)$$

with $y_0 = x_{j-\frac{1}{2}}$, $y_m = x_{j+\frac{1}{2}}$. The scheme for the cell average is then rewritten as

$$\begin{aligned}
\bar{u}_j^{n+1} &= \omega_m \left[u^-_{j+\frac{1}{2}} - \frac{\lambda}{\omega_m} \left(h(u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) - h(u^+_{j-\frac{1}{2}}, u^-_{j+\frac{1}{2}}) \right) \right] \\
&\quad + \omega_0 \left[u^+_{j-\frac{1}{2}} - \frac{\lambda}{\omega_0} \left(h(u^+_{j-\frac{1}{2}}, u^-_{j+\frac{1}{2}}) - h(u^-_{j-\frac{1}{2}}, u^+_{j-\frac{1}{2}}) \right) \right] \\
&\quad + \sum_{\ell=1}^{m-1} \omega_\ell p_j(y_\ell) \\
&= \omega_m H_{\lambda/\omega_m}(u^+_{j-\frac{1}{2}}, u^-_{j+\frac{1}{2}}, u^+_{j+\frac{1}{2}}) + \omega_0 H_{\lambda/\omega_0}(u^-_{j-\frac{1}{2}}, u^+_{j-\frac{1}{2}}, u^-_{j+\frac{1}{2}}) \\
&\quad + \sum_{\ell=1}^{m-1} \omega_\ell p_j(y_\ell).
\end{aligned}$$

Therefore, if

$$m \leq p_j(y_\ell) \leq M$$

at all Legendre Gauss-Lobatto quadrature points and a reduced CFL condition

$$\lambda/\omega_m = \lambda/\omega_0 \leq \lambda_0$$

is satisfied, then

$$m \leq \bar{u}_j^{n+1} \leq M.$$

2. The second difficulty is: given

$$m \le \bar{u}_j^{n+1} \le M, \qquad \forall j$$

how to obtain an accurate reconstruction $u^{n+1}(x)$ which satisfy

$$m \le u^{n+1}(x) \le M, \qquad \forall x.$$

Previous work was mainly for relatively lower order schemes (second or third order accurate), and would typically require an evaluation of the extrema of $u^{n+1}(x)$, which, for a piecewise polynomial of higher degree, is quite costly.

We have figured out a way to obtain such reconstruction with a very simple scaling limiter, which only requires the evaluation of $u^{n+1}(x)$ at certain pre-determined quadrature points and does not destroy accuracy:

We replace $p_j(x)$ by the limited polynomial $\tilde{p}_j(x)$ defined by

$$\tilde{p}_j(x) = \theta_j(p_j(x) - \bar{u}_j^n) + \bar{u}_j^n$$

where

$$\theta_j = \min\left\{ \left| \frac{M - \bar{u}_j^n}{M_j - \bar{u}_j^n} \right|, \left| \frac{m - \bar{u}_j^n}{m_j - \bar{u}_j^n} \right|, 1 \right\},$$

with

$$M_j = \max_{x \in S_j} p_j(x), \qquad m_j = \min_{x \in S_j} p_j(x)$$

where S_j is the set of Legendre Gauss-Lobatto quadrature points of cell I_j.

Clearly, this limiter is just a simple scaling of the original polynomial around its average.

The following lemma, guaranteeing the maintenance of accuracy of this simple limiter, is proved in Zhang and Shu, JCP 2010a:

Lemma: Assume $\bar{u}_j^n \in [m, M]$ and $p_j(x)$ is an $O(\Delta x^p)$ approximation, then $\tilde{p}_j(x)$ is also an $O(\Delta x^p)$ approximation.

3. The third difficulty is how to generalize the algorithm and result to 2D (or higher dimensions). Algorithms which would require an evaluation of the extrema of the reconstructed polynomials $u^{n+1}(x, y)$ would not be easy to generalize at all.

   Our algorithm uses only explicit Euler forward or SSP (also called TVD) Runge-Kutta or multi-step time discretizations, and a simple scaling limiter involving just evaluation of the polynomial at certain quadrature points, hence easily generalizes to 2D or higher dimensions on structured or unstructured meshes, with strict maximum-principle-satisfying property and provable high order accuracy.

The technique has been generalized to the following situations maintaining uniformly high order accuracy:

- 2D scalar conservation laws on rectangular or triangular meshes with strict maximum principle (Zhang and Shu, JCP 2010a; Zhang, Xia and Shu, JSC 2012).

- 2D incompressible equations in the vorticity-streamfunction formulation (with strict maximum principle for the vorticity), and 2D passive convections in a divergence-free velocity field, i.e.

$$\omega_t + (u\omega)_x + (v\omega)_x = 0,$$

  with a given divergence-free velocity field (u, v), again with strict maximum principle (Zhang and Shu, JCP 2010a; Zhang, Xia and Shu, JSC 2012).

Positivity-preserving for systems

The framework of establishing maximum-principle-satisfying schemes for scalar equations can be generalized to hyperbolic systems to preserve the positivity of certain physical quantities, such as density and pressure of compressible gas dynamics.

Compressible Euler equations:

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

with

$$u = \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix}, \qquad f(u) = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix},$$

where $E = e + \frac{1}{2}\rho v^2$. The internal energy e is related to density and pressure through the equation of states (EOS). For the ideal gas, we have $e = \frac{p}{\gamma - 1}$ with $\gamma = 1.4$ for air.

The main ingredients for designing positivity-preserving schemes for systems are:

- A first order explicit scheme which can keep the positivity of the desired quantities (e.g. density and pressure) under a suitable CFL condition.

  Examples include the Godunov scheme, Lax-Friedrichs scheme, kinetic scheme, HLLC scheme, etc.

- The quantity for which positivity is desired is one of the components of the conserved variable u (for example the density ρ), or is a concave function of the conserved variable u (for example the pressure p or the internal energy e). Under this assumption, the region of positivity of the desired quantities is a convex region in the u space.

With these ingredients, the technique to enforce maximum-principle for scalar equations can be directly generalized to enforce positivity of the desired quantities without affecting the high order accuracy of the finite volume or DG schemes.

Positivity-preserving finite volume or DG schemes have been designed for:

- One and multi-dimensional compressible Euler equations maintaining positivity of density and pressure (Zhang and Shu, JCP 2010b; Zhang, Xia and Shu, JSC 2012).
- One and two-dimensional shallow water equations maintaining non-negativity of water height and well-balancedness for problems with dry areas (Xing, Zhang and Shu, Advances in Water Resources 2010; Xing and Shu, Advances in Water Resources 2011).

- One and multi-dimensional compressible Euler equations with source terms (geometric, gravity, chemical reaction, radiative cooling) maintaining positivity of density and pressure (Zhang and Shu, JCP 2011).

- One and multi-dimensional compressible Euler equations with gaseous detonations maintaining positivity of density, pressure and reactant mass fraction, with a new and simplified implementation of the pressure limiter. DG computations are stable without using the TVB limiter (Wang, Zhang, Shu and Ning, JCP 2012).

- A minimum entropy principle satisfying high order scheme for gas dynamics equations (Zhang and Shu, Num Math 2012).

Other generalizations

- Positivity-preserving high order finite difference WENO schemes for compressible Euler equations (Zhang and Shu, JCP 2012).
- Simplified version for WENO finite volume schemes without the need to evaluate solutions at quadrature points inside the cell (Zhang and Shu, Proceedings of the Royal Society A, 2011).

- Positivity-preserving for PDEs involving global integral terms including a hierarchical size-structured population model (Zhang, Zhang and Shu, JCAM 2011) and Vlasov-Boltzmann transport equations (Cheng, Gamba and Proft, Math Comp, 2012).
- Positivity-preserving semi-Lagrangian schemes (Qiu and Shu, JCP 2011; Rossmanith and Seal, JCP 2011).
- Positivity-preserving first order and higher order Lagrangian schemes for multi-material flows (Cheng and Shu, JCP submitted).

Numerical results

Example 1. Accuracy check. For the incompressible Euler equation in the vorticity-streamfunction formulation, with periodic boundary condition and initial data $\omega(x, y, 0) = -2\sin(x)\sin(y)$ on the domain $[0, 2\pi] \times [0, 2\pi]$, the exact solution is $\omega(x, y, t) = -2\sin(x)\sin(y)$. We clearly observe the designed order of accuracy for this solution.

Table 1: Incompressible Euler equations. P^2 for vorticity, $t = 0.5$.

N×N	L^1 error	order	L^∞ error	order
16×16	5.12E-4	–	1.40E-3	–
32×32	3.75E-5	3.77	1.99E-4	2.81
64×64	3.16E-6	3.57	2.74E-5	2.86
128×128	2.76E-7	3.51	3.56E-6	2.94

Example 2. The vortex patch problem. We solve the incompressible Euler equations in $[0,2\pi]\times[0,2\pi]$ with the initial condition

$$\omega(x,y,0)=\begin{cases} -1, & \frac{\pi}{2}\le x\le\frac{3\pi}{2}, \frac{\pi}{4}\le y\le\frac{3\pi}{4}; \\ 1, & \frac{\pi}{2}\le x\le\frac{3\pi}{2}, \frac{5\pi}{4}\le y\le\frac{7\pi}{4}; \\ 0, & \text{otherwise} \end{cases}$$

and periodic boundary conditions. The contour plots of the vorticity ω are given for $t=10$. Again, we cannot observe any significant difference between the two results in the contour plots. The cut along the diagonal gives us a clearer view of the advantage in using the limiter.

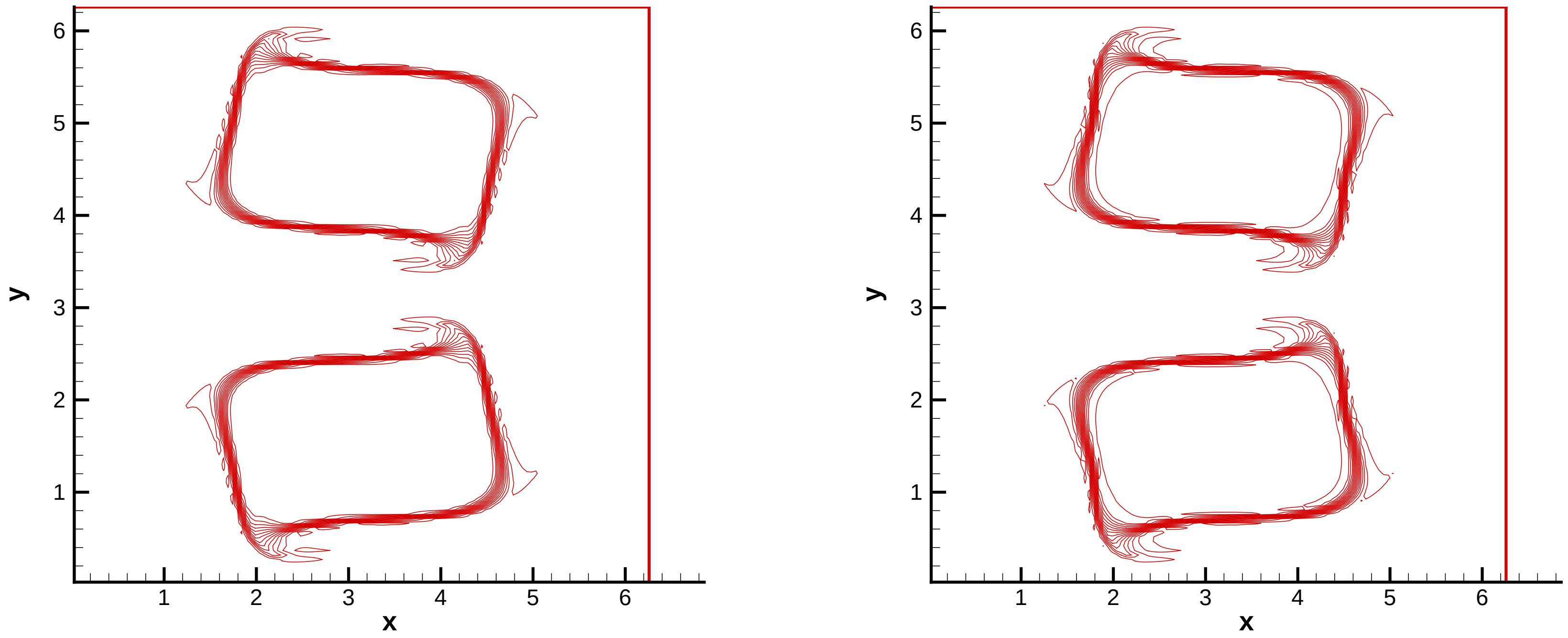


Figure 13: Vorticity at $t = 10$, P^2. 30 equally spaced contours from -1.1 to 1.1. 128^2 mesh. Left: with limiter; Right: without limiter.

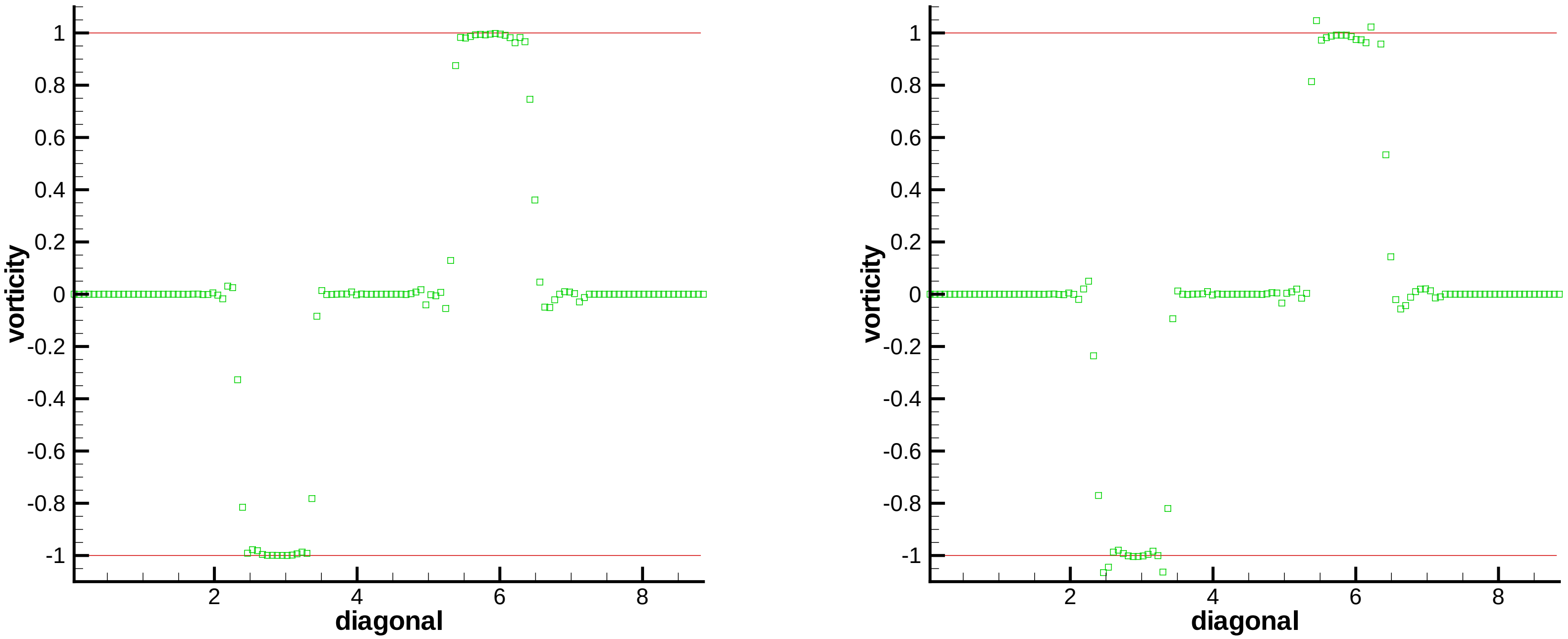


Figure 14: Vorticity at $t = 10$, P^2. Cut along the diagonal. 128^2 mesh. Left: with limiter; Right: without limiter.

Example 3. The Sedov point-blast wave in one dimension. For the initial condition, the density is 1, velocity is zero, total energy is 10^{-12} everywhere except that the energy in the center cell is the constant $\frac{E_0}{\Delta x}$ with $E_0 = 3200000$ (emulating a δ-function at the center). $\gamma = 1.4$. The computational results are shown in Figure 15. We can see the shock is captured very well.

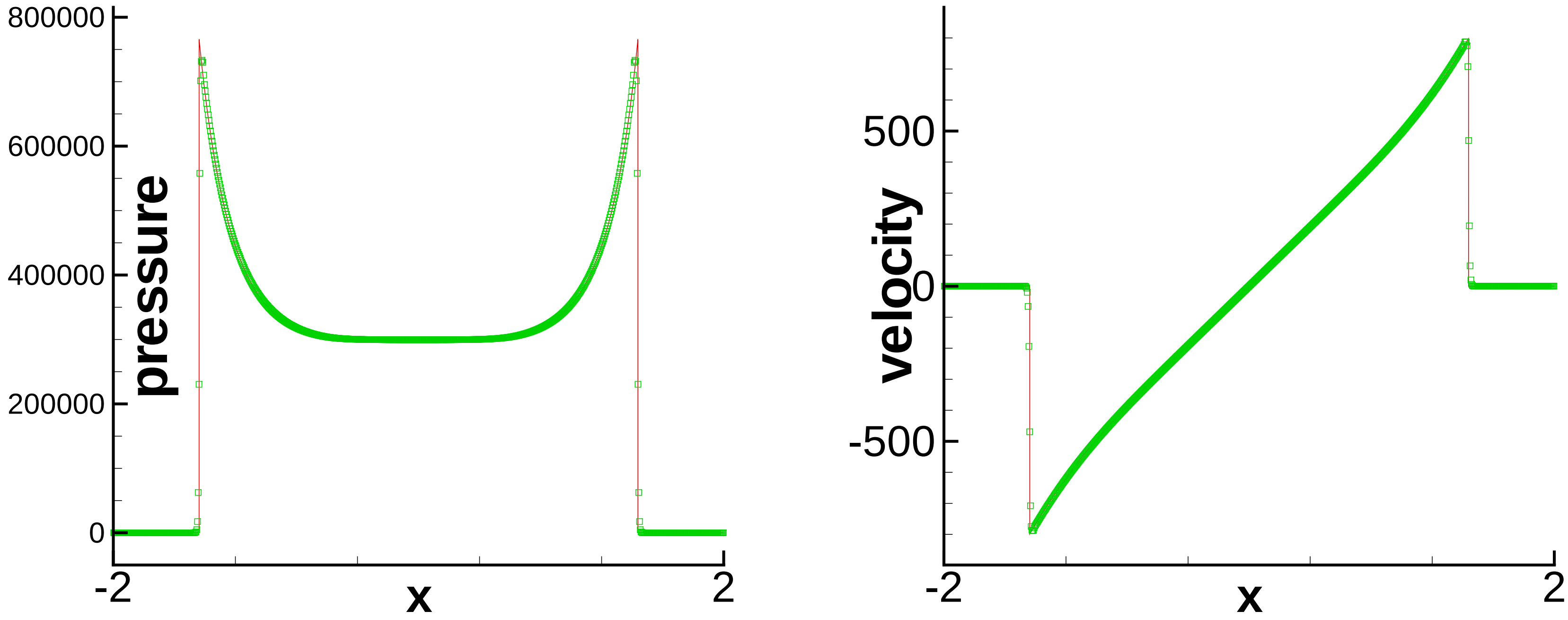


Figure 15: 1D Sedov blast. The solid line is the exact solution. Symbols are numerical solutions. $T = 0.001$. $N = 800$. $\Delta x = \frac{4}{N}$. TVB limiter parameters $(M_1, M_2, M_3) = (15000, 20000, 15000)$. Pressure (left) and velocity (right).

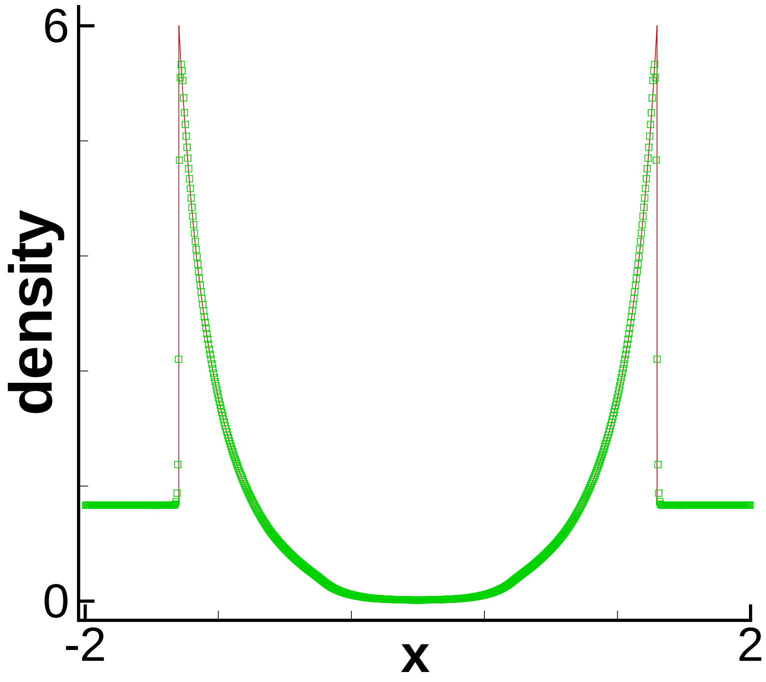


Figure 16: 1D Sedov blast. The solid line is the exact solution. Symbols are numerical solutions. $T = 0.001$. $N = 800$. $\Delta x = \frac{4}{N}$. TVB limiter parameters $(M_1, M_2, M_3) = (15000, 20000, 15000)$. Density.

Example 4. The Sedov point-blast wave in two dimensions. The computational domain is a square. For the initial condition, the density is 1, velocity is zero, total energy is 10^{-12} everywhere except that the energy in the lower left corner cell is the constant $\frac{0.244816}{\Delta x \Delta y}$. $\gamma = 1.4$. See Figure 17. The computational result is comparable to those in the literature, e.g. those computed by Lagrangian methods.

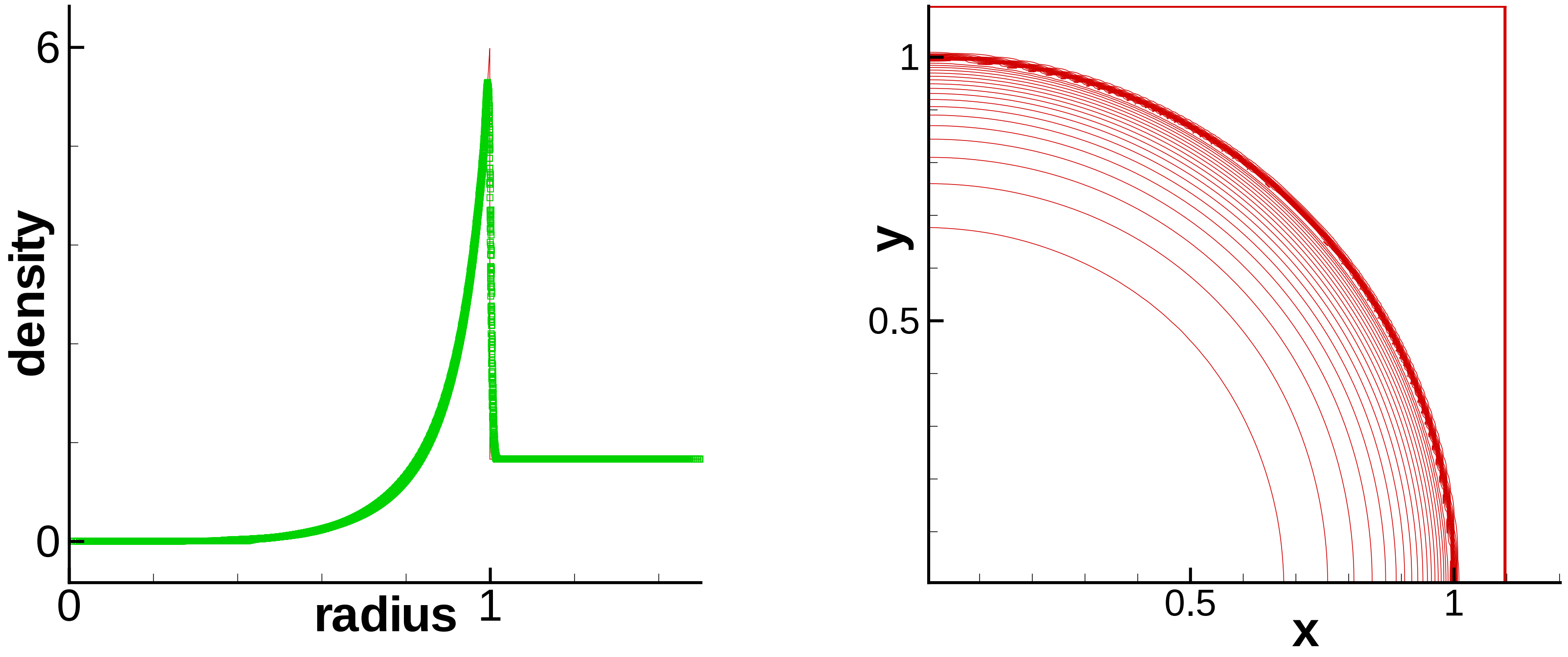


Figure 17: 2D Sedov blast, plot of density. $T = 1$. $N = 160$. $\Delta x = \Delta y = \frac{1.1}{N}$. TVB limiter parameters $(M_1, M_2, M_3, M_4) = (8000, 16000, 16000, 8000)$.

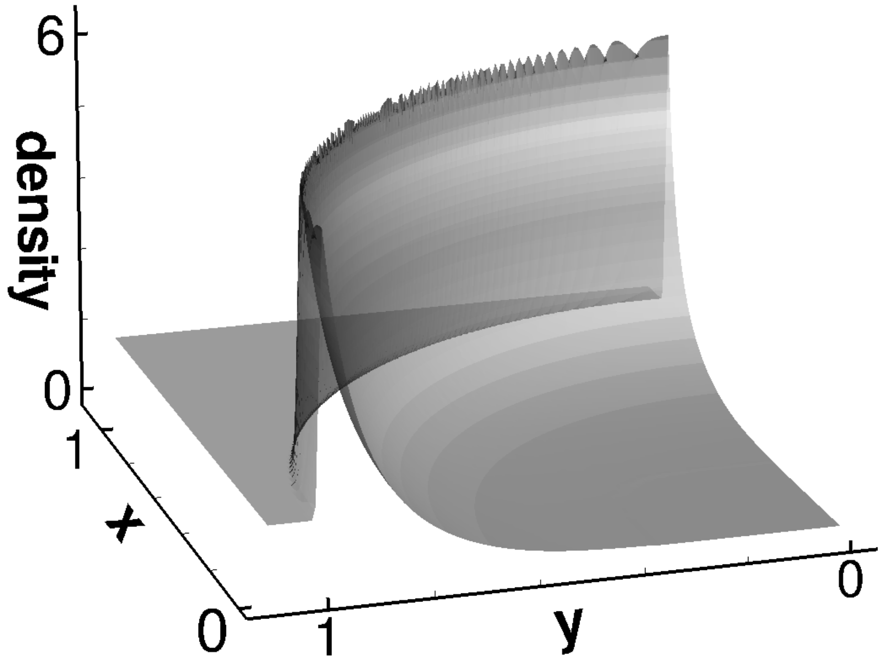


Figure 18: 2D Sedov blast, plot of density. $T = 1$. $N = 160$. $\Delta x = \Delta y = \frac{1.1}{N}$. TVB limiter parameters $(M_1, M_2, M_3, M_4) = (8000, 16000, 16000, 8000)$.

Example 5. We consider two Riemann problems. The first one is a double rarefaction. We did two tests, one is a one-dimensional double rarefaction, for which the initial condition is $\rho_L = \rho_R = 7, u_L = -1, u_R = 1$, $p_L = p_R = 0.2$ and $\gamma = 1.4$. The other one is a two-dimensional double rarefaction with the initial condition $\rho_L = \rho_R = 7, u_L = -1, u_R = 1$, $v_L = v_R = 0, p_L = p_R = 0.2$. The exact solution contains vacuum.

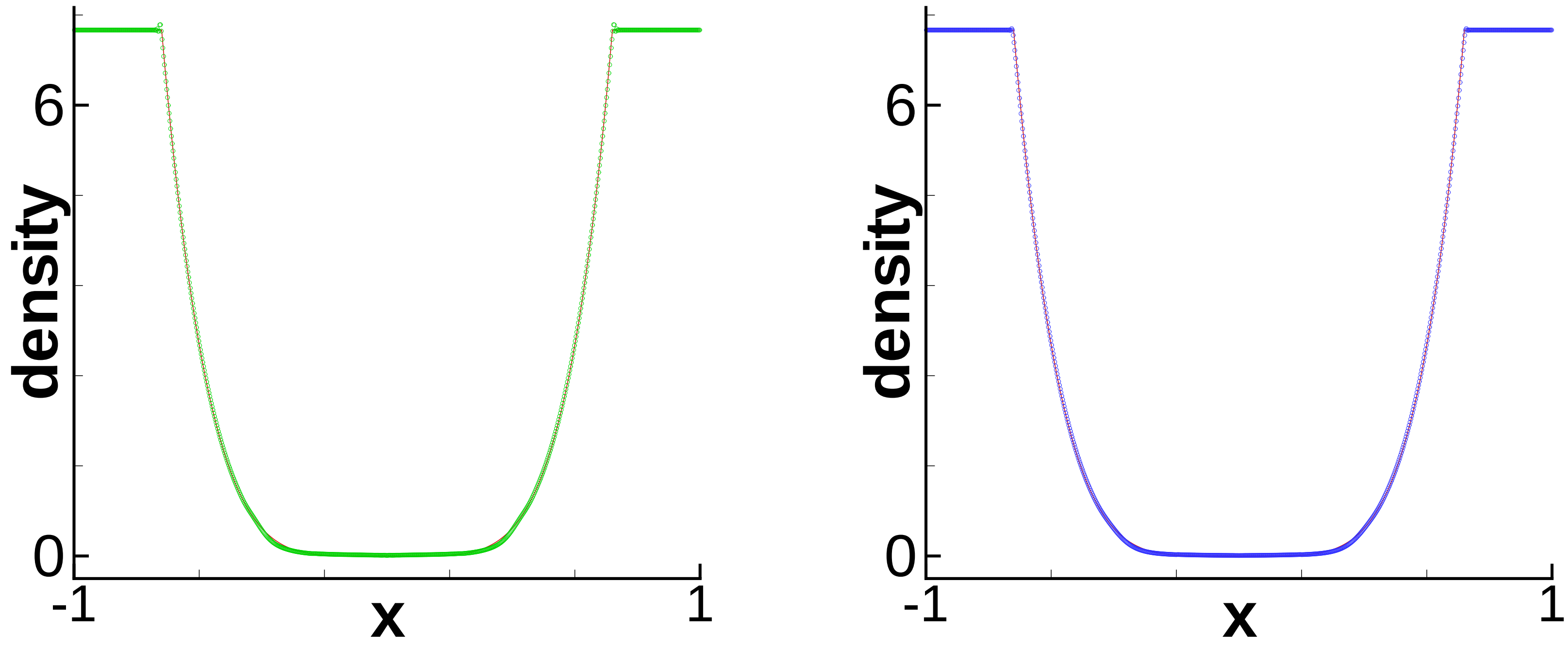


Figure 19: Double rarefaction problem. T=0.6. Left: 1D problem. Right: Cut at $y = 0$ for the 2D problem. Every fourth cell is plotted. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{2}{N}$, $N = 800$ with the positivity limiter. Density.

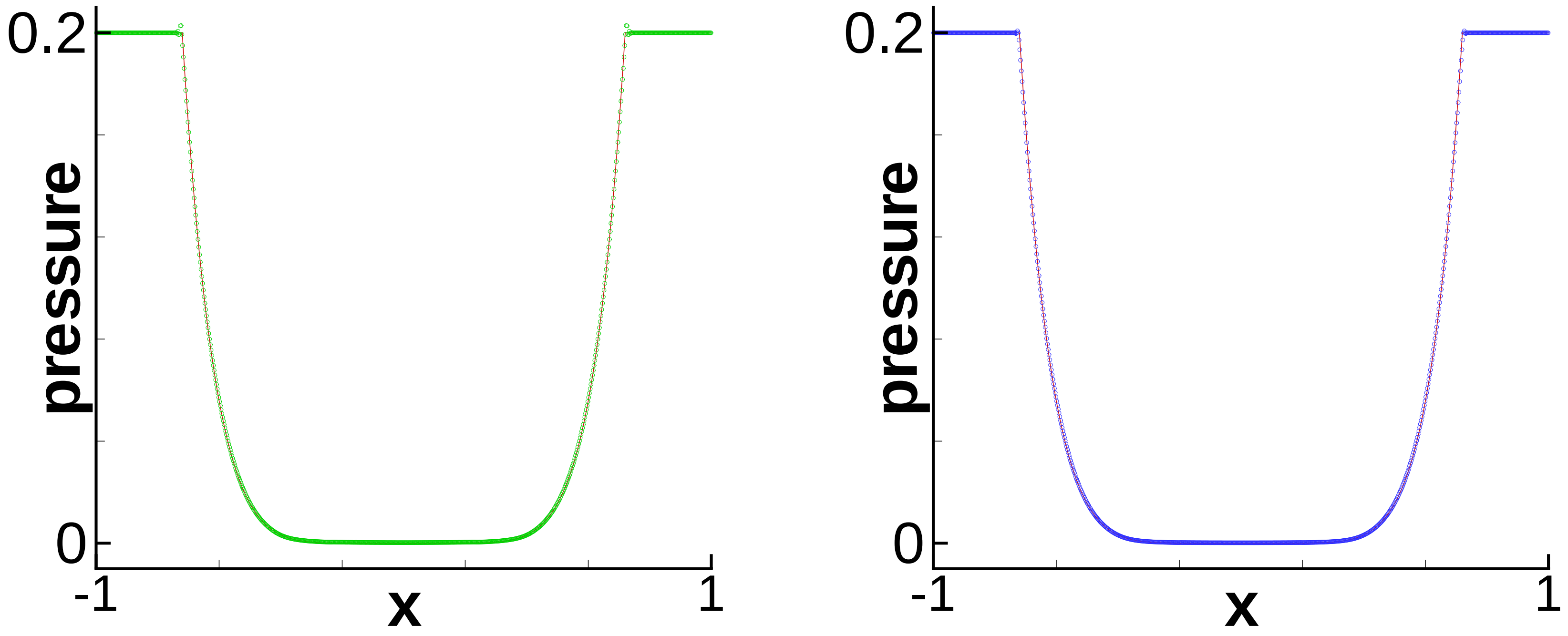


Figure 20: Double rarefaction problem. T=0.6. Left: 1D problem. Right: Cut at $y = 0$ for the 2D problem. Every fourth cell is plotted. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{2}{N}$, $N = 800$ with the positivity limiter. Pressure.

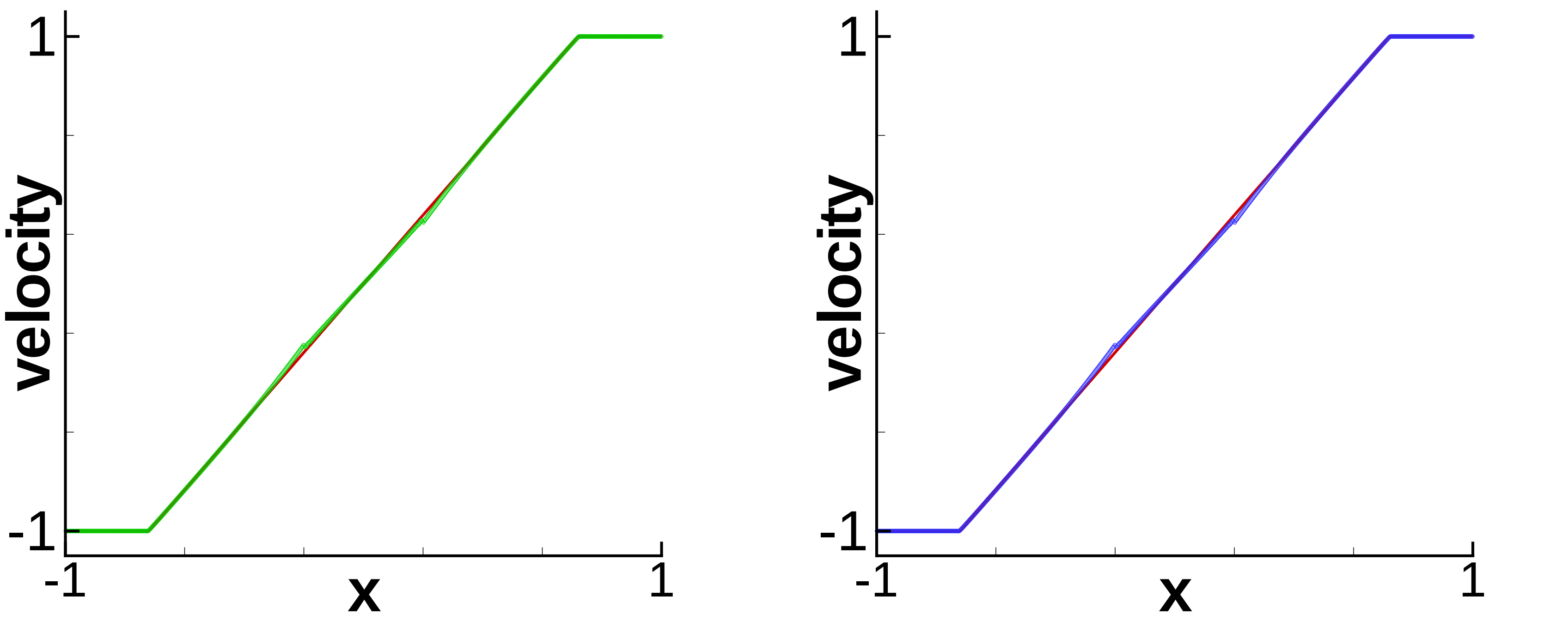


Figure 21: Double rarefaction problem. T=0.6. Left: 1D problem. Right: Cut at $y = 0$ for the 2D problem. Every fourth cell is plotted. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{2}{N}$, $N = 800$ with the positivity limiter. Velocity.

The second one is a 1D Leblanc shock tube problem. The initial condition is $\rho_L = 2$, $\rho_R = 0.001$, $u_L = u_R = 0$, $p_L = 10^9$, $p_R = 1$, and $\gamma = 1.4$. See the next figure for the results of 800 cells and 6400 cells.

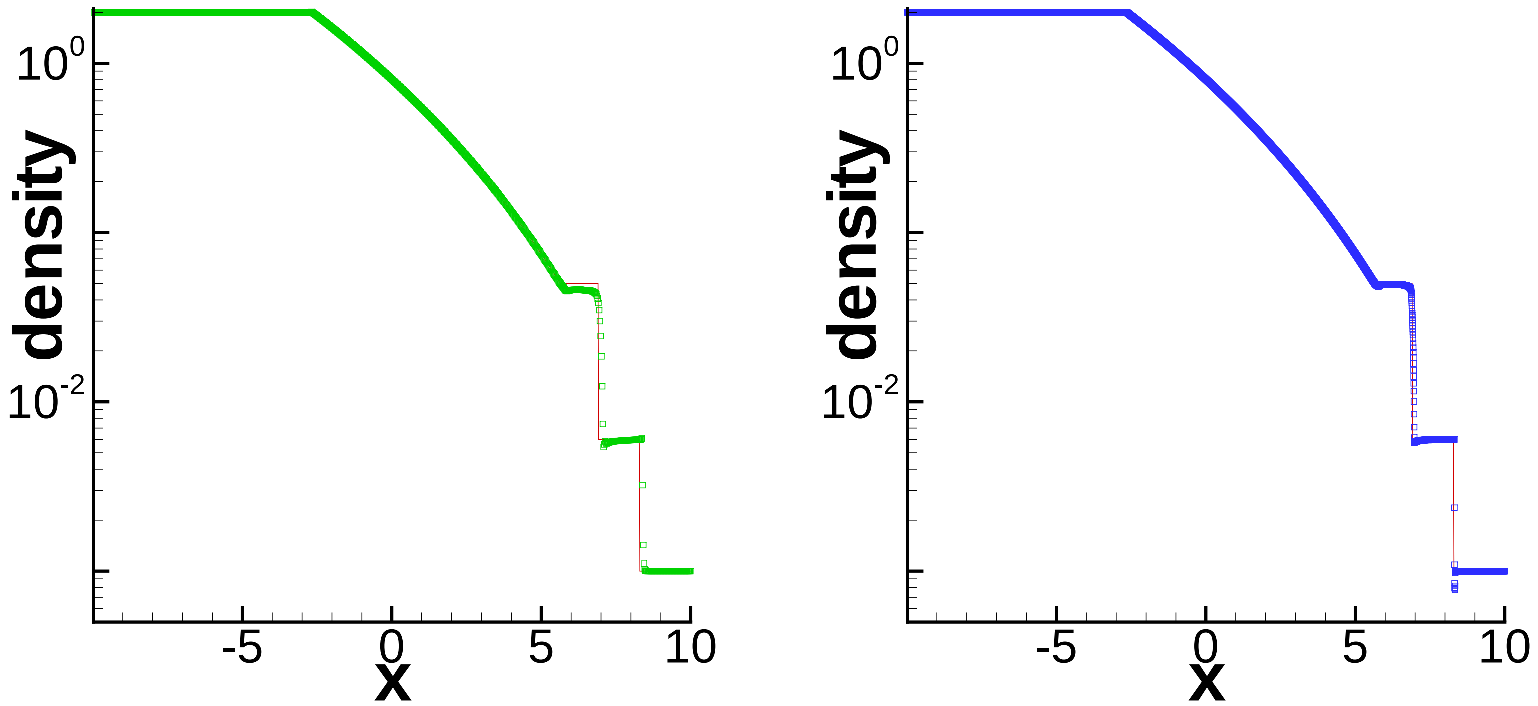


Figure 22: Leblanc problem. $T = 0.0001$. Left: $N = 800$. Right: $N = 6400$. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{20}{N}$ with the positivity limiter. $\log$-scale of density.

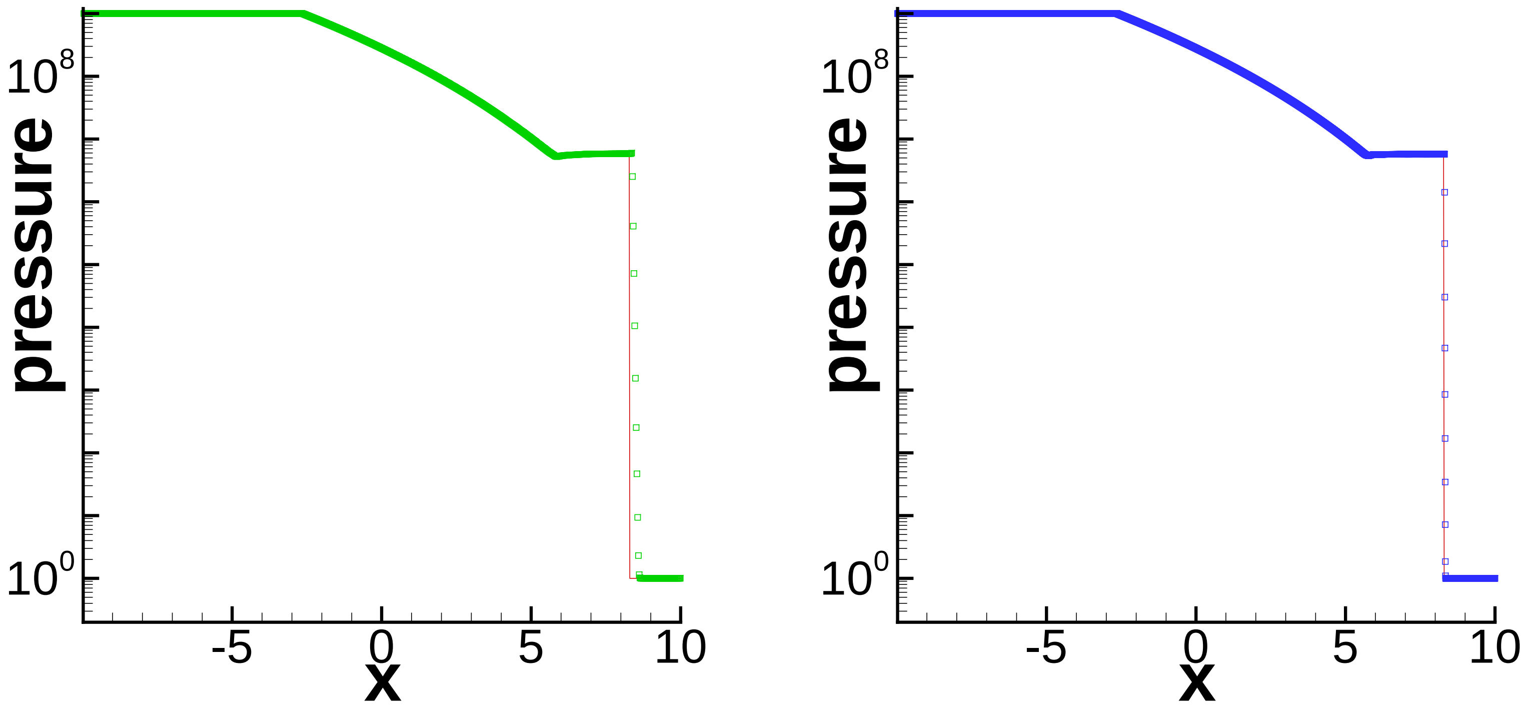


Figure 23: Leblanc problem. $T = 0.0001$. Left: $N = 800$. Right: $N = 6400$. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{20}{N}$ with the positivity limiter. $\log$-scale of pressure.

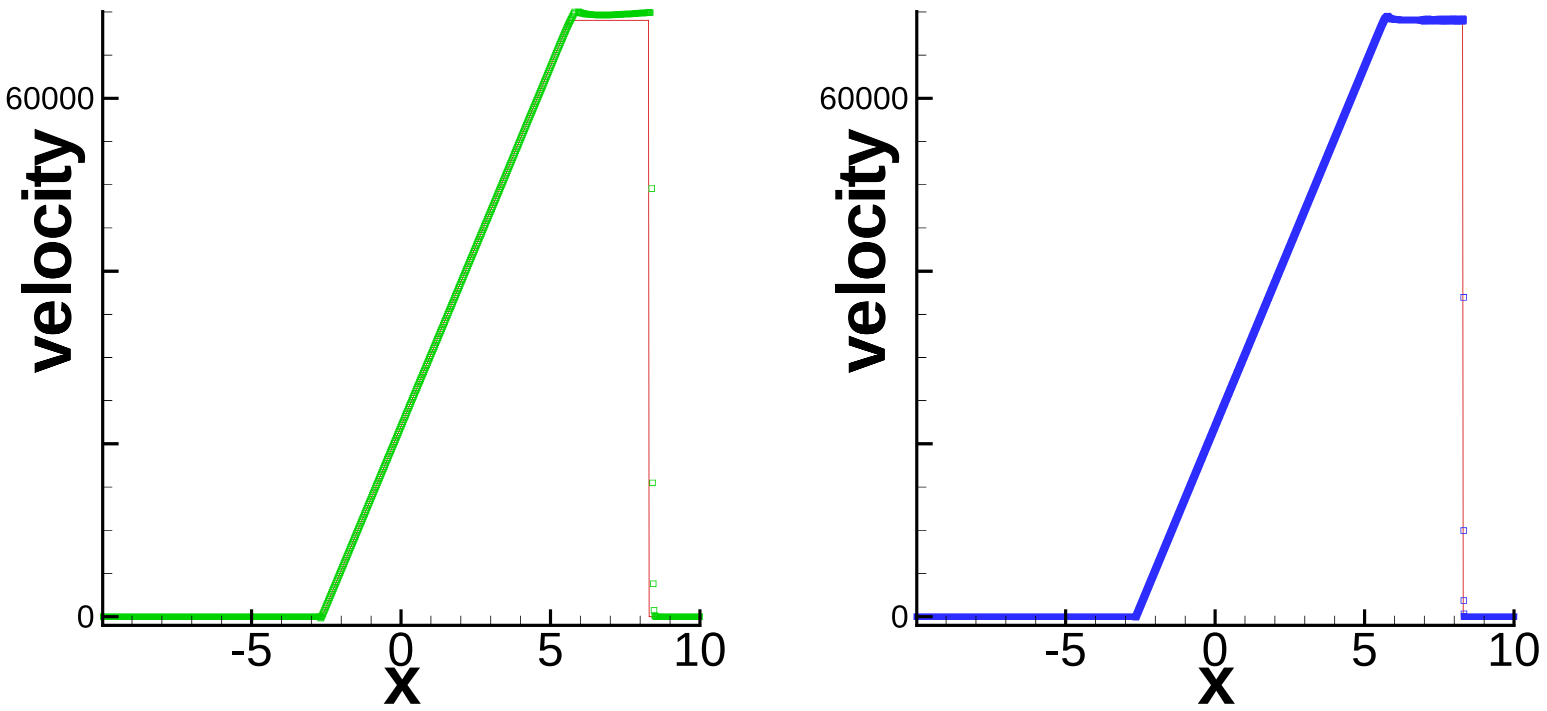


Figure 24: Leblanc problem. $T = 0.0001$. Left: $N = 800$. Right: $N = 6400$. The solid line is the exact solution. Symbols are numerical solutions. $\Delta x = \frac{20}{N}$ with the positivity limiter. Velocity.

Example 6. To simulate the gas flows and shock wave patterns which are revealed by the Hubble Space Telescope images, one can implement theoretical models in a gas dynamics simulator. The two-dimensional model without radiative cooling is governed by the compressible Euler equations. The velocity of the gas flow is extremely high, and the Mach number could be hundreds or thousands. A big challenge for computation is, even for a state-of-the-art high order scheme, negative pressure could appear since the internal energy is very small compared to the huge kinetic energy (Ha, Gardner, Gelb and Shu, JSC 2005).

First, we compute a Mach 80 (i.e. the Mach number of the jet inflow is 80 with respect to the soundspeed in the jet gas) problem without the radiative cooling.

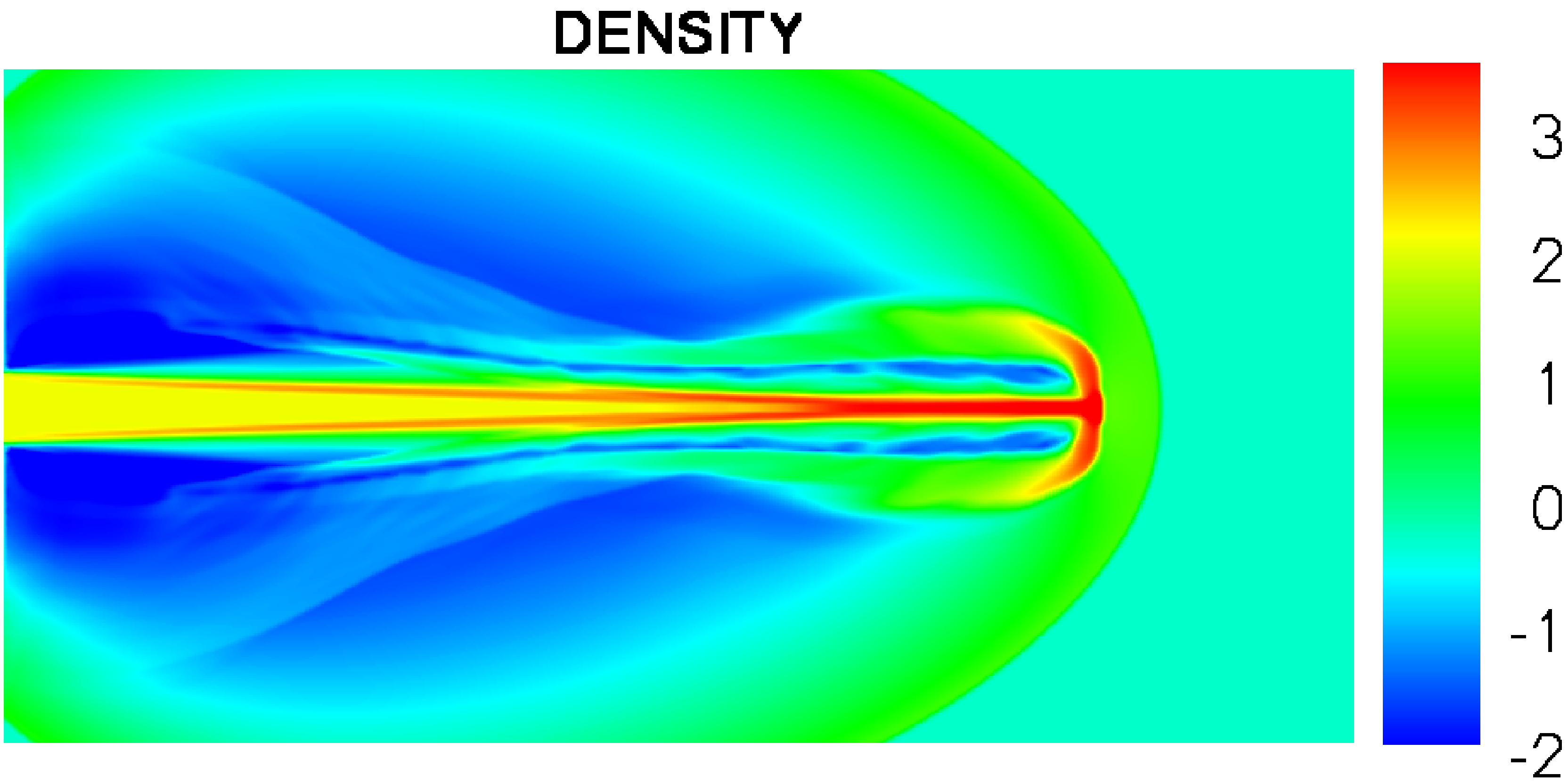


Figure 25: Simulation of Mach 80 jet without radiative cooling. Scales are logarithmic. Density.

Second, to demonstrate the robustness of our method, we compute a Mach 2000 problem. The domain is $[0,1]\times[0,0.5]$. The width of the jet is 0.1. The terminal time is 0.001. The speed of the jet is 800, which is around Mach 2100 with respect to the soundspeed in the jet gas. The computation is performed on a 640×320 mesh. TVB limiter parameters are $M_1=M_2=M_3=M_4=10000000$.

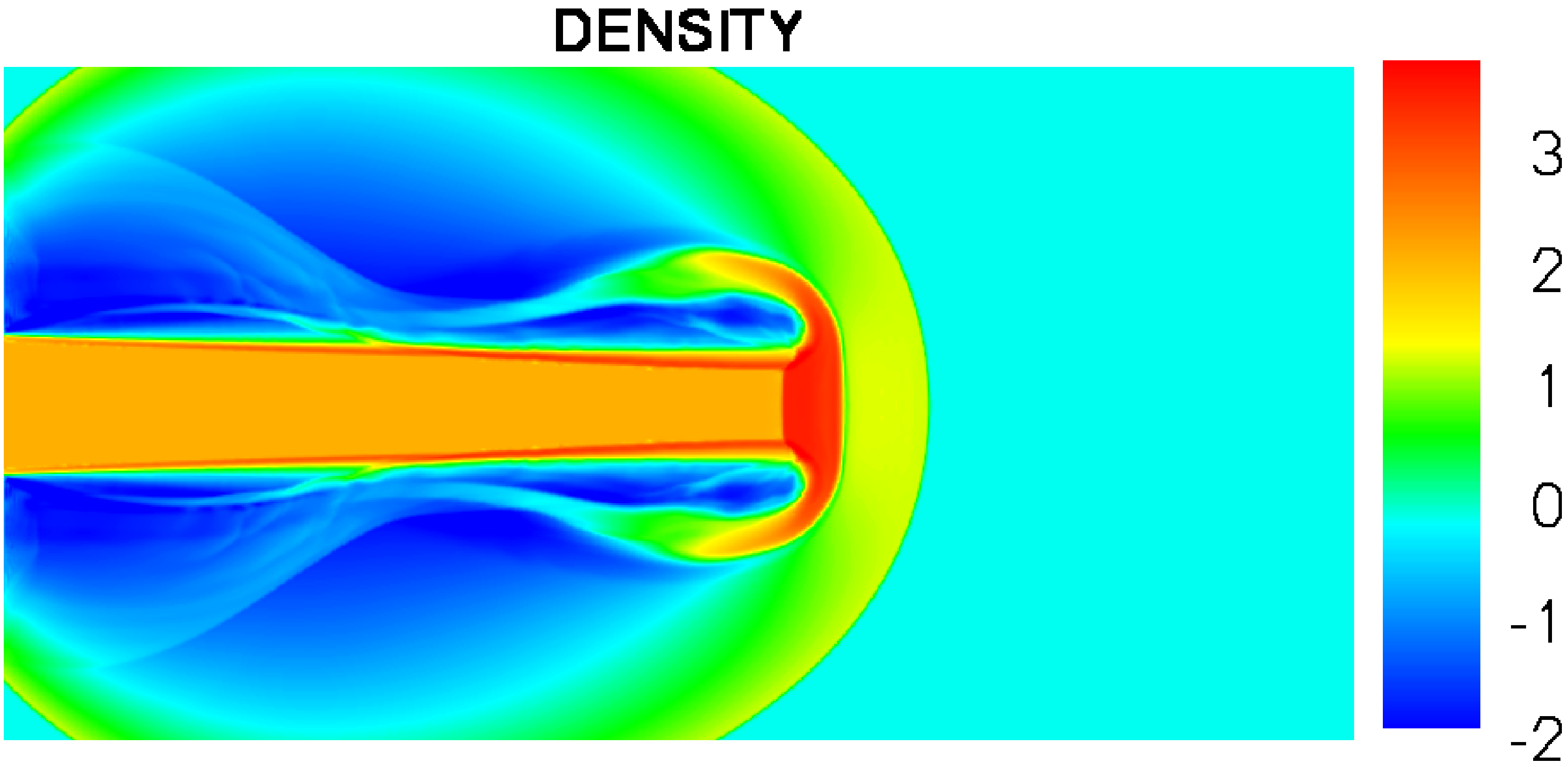


Figure 26: Simulation of Mach 2000 jet without radiative cooling. Scales are logarithmic. Density.

Lastly, we compute a Mach 80 (i.e. the Mach number of the jet inflow is 80 with respect to the soundspeed in the jet gas) problem with the radiative cooling to test the positivity-preserving property with the radiative cooling source term.

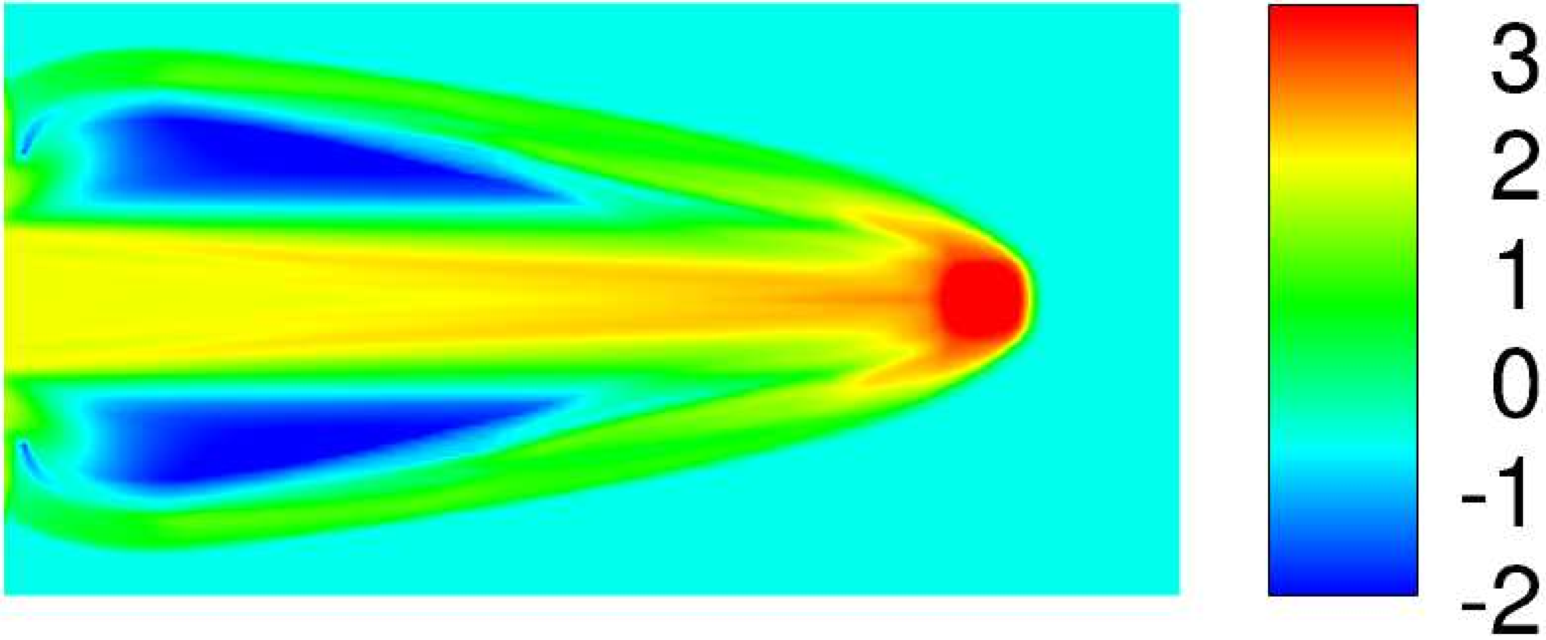


Figure 27: Simulation of Mach 80 jet with radiative cooling. The third order positivity-preserving RKDG scheme with the TVB limiter. Scales are logarithmic. Density.

Example 7. Shock diffraction problem. Shock passing a backward facing corner of 135°. It is easy to get negative density and/or pressure below the corner. This problem also involves mixed triangular / rectangular meshes for the DG method. The initial conditions are, if $x < 1.5$ and $y \geq 2$, $(\rho, u, v, E, Y) = (11, 6.18, 0, 970, 1)$; otherwise, $(\rho, u, v, E, Y) = (1, 0, 0, 55, 1)$. The boundary conditions are reflective. The terminal time is $t = 0.68$.

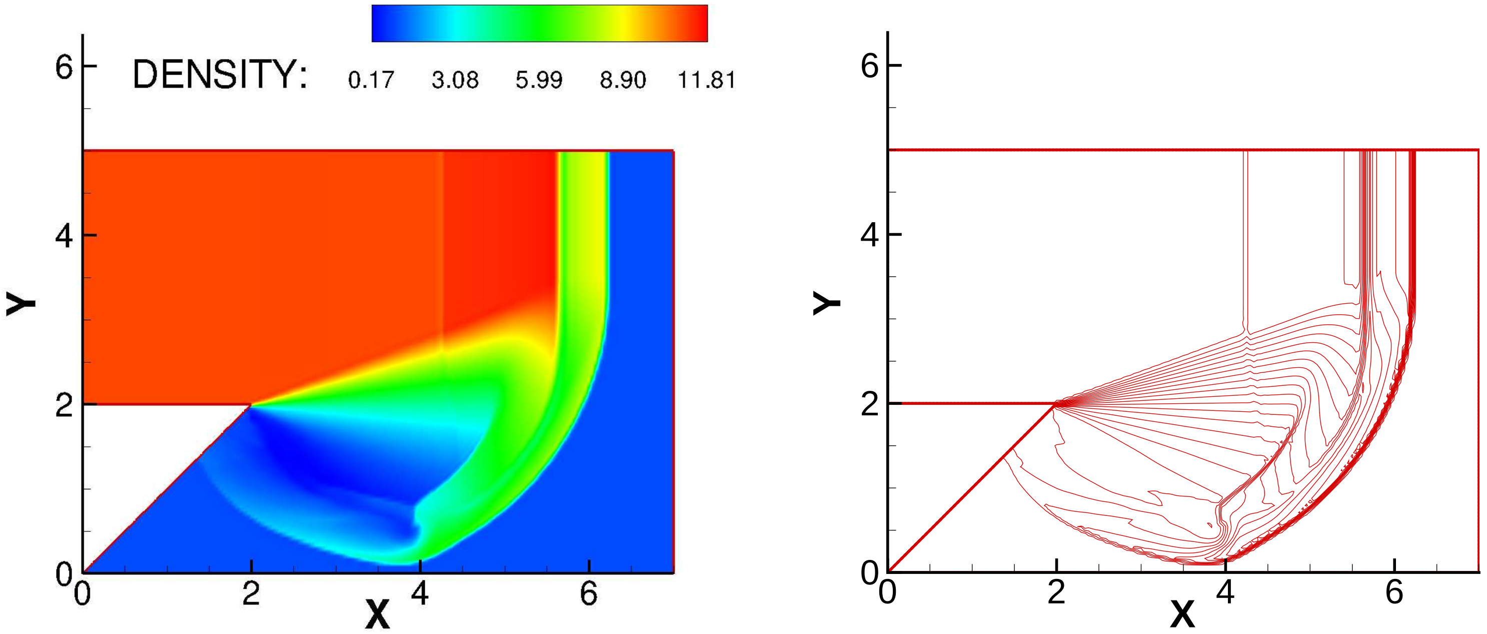


Figure 28: Density. Detonation diffraction at a 135° corner.

DG method for problems involving δ-singularities

DG method for discontinuous solutions

Even though the major motivation to design the DG method for solving hyperbolic equations is to resolve discontinuous solutions more effectively, there are not many convergence and error estimate results for DG method with discontinuous solutions.

Previous work on this issue:

- Johnson et al CMAME 1984; Johnson and Pitkäranta Math Comp 1986; Johnson et al Math Comp 1987: error estimates for piecewise linear streamline diffusion and DG methods for stationary (or space-time) hyperbolic equations. Pollution region around discontinuity: $\mathcal{O}(h^{1/2}\log(1/h))$.

- Cockburn and Guzmán SINUM 2008: RKDG2 (second order in space and time), Pollution region around discontinuity: $\mathcal{O}(h^{1/2}\log(1/h))$ on the downwind side and $\mathcal{O}(h^{2/3}\log(1/h))$ on the upwind side. The result is for uniform meshes and does not hold when the CFL number goes to zero, or for the semi-discrete DG scheme. Also it does not generalize easily to higher order either in space or in time.

In Zhang and Shu, Num Math to appear, we investigate the RKDG3 scheme with arbitrary polynomial degree $k \geq 1$ in space and third order TVD Runge-Kutta in time, on arbitrary quasi-uniform mesh, for solving the linear model equation

$$\begin{aligned} u_t + \beta u_x &= 0 \\ u(x,0) = u^0(x) & \end{aligned}$$

where β is a constant, $u_0(x)$ has compact support, has a sole discontinuity at $x = 0$ and is sufficiently smooth everywhere else.

We prove the following error estimate:

Theorem: Assume the CFL number $\lambda := |\beta|\Delta t/h_{\min}$ is small enough, there holds

$$\|u(t^N) - u_h^N\|_{L^2(\mathbb{R}\backslash\mathcal{R}_T)} \leq M(h^{k+1} + \Delta t^3), \tag{14}$$

where $M > 0$ is independent of h and Δt, but may depend on the final time T, the norm of the exact solution in smooth regions, and the jump at the discontinuity point. Here $\mathcal{R}_T$ is the pollution region at the final time T, given by

$$\mathcal{R}_T = (\beta T - C\sqrt{T\beta\nu^{-1}}h^{1/2}\log(1/h), \beta T + C\sqrt{T\beta\nu^{-1}}h^{1/2}\log(1/h)), \tag{15}$$

where $C > 0$ is independent of $\nu = h_{\min}/h_{\max}, \lambda, \beta, h, \Delta t$ and T.

Numerical results:

We use uniform spatial meshes, together with the uniform time stepping with the CFL number $\lambda = 0.18$. We compute the errors and convergence orders at the final time $T = 0.25$, on the left and the right, respectively, of the singularity $x = 0.625$, namely,

$$\mathcal{R}_T^L = (-\infty, 0.625 - 0.5h^{1/2}), \quad \text{and} \quad \mathcal{R}_T^R = (0.625 + 0.8h^{1/2}, +\infty). \tag{16}$$

The errors and convergence orders for $k = 2$ are listed in Table 2. As we can see, the optimal orders of convergence are realized; this confirms the prediction of our theorem that the pollution region sizes on both sides of the discontinuity are no larger than about the order $\mathcal{O}(h^{1/2})$.

Table 2: Errors and convergence orders in the L^2-norm and maximum norm, to the left and to the right of the singularity. Here $k = 2$ and $\lambda = 0.18$.

	Left-hand interval $\mathcal{R}_T^L$				Right-hand interval $\mathcal{R}_T^R$			
$1/h_\ell$	L^2-err	order	L^∞-err	order	L^2-err	order	L^∞-err	order
1000	5.90e-8		8.36e-7		4.75e-8		8.21e-7	
2000	7.90e-9	2.90	1.04e-7	3.01	7.03e-9	2.76	1.03e-7	3.00
4000	1.03e-9	2.94	1.30e-8	3.00	9.61e-10	2.87	1.29e-8	3.00
8000	1.33e-10	2.96	1.63e-9	3.00	1.27e-10	2.92	1.61e-9	3.00
16000	1.70e-11	2.97	2.04e-10	3.00	1.65e-11	2.95	2.02e-10	3.00
32000	2.15e-12	2.98	2.55e-11	3.00	2.10e-12	2.97	2.53e-11	3.00

Now we use the piecewise cubic polynomials ($k = 3$) on uniform meshes, as the finite element space in the RKDG3 method. We also compute the solution until the same final time $T = 0.25$, To obtain the optimal fourth order accuracy, we take the time step $\Delta t = 0.18h^{4/3}$, where h is the uniform mesh length. The errors and convergence orders on the two domains same as (16), are listed in Table 3. We can observe that, the optimal orders of convergence is also achieved.

Table 3: Errors and convergence orders in the L^2-norm and maximum norm, to the left and to the right of the singularity. Here $k = 3$.

	Left-hand interval $\mathcal{R}_T^L$				Right-hand interval $\mathcal{R}_T^R$			
$1/h_\ell$	L^2-err	order	L^∞-err	order	L^2-err	order	L^∞-err	order
1000	1.58e-10		2.00e-9		1.57e-10		2.00e-9	
2000	1.05e-10	3.91	1.25e-10	4.00	9.76e-12	4.01	1.25e-10	4.00
4000	7.03e-13	3.90	9.42e-12	3.73	6.35e-13	3.94	7.82e-12	4.00
8000	4.67e-14	3.91	6.52e-13	3.85	4.27e-14	3.89	5.48e-13	3.84

Several questions to be addressed:

- Is the bound on the pollution region $\mathcal{O}(h^{1/2}\log(1/h))$ sharp, especially considering that Cockburn and Guzmán SINUM 2008 has a sharper bound $\mathcal{O}(h^{2/3}\log(1/h))$ on the upwind side for RKDG2, and that it is expected in the numerical hyperbolic community (Harten, without formal proof) that higher order schemes should have a narrower pollution region.

Using a carefully designed numerical least square procedure over several mesh refinements, we can estimate the boundary of the pollution region (beyond which the normal $O(h^{k+1})$ convergence is achieved).

For $k = 2$ (third order RKDG in space and time), the result, showing in the next figure, clearly indicates that the least square process provides $s = 0.490$ for the left side and $s = 0.522$ for the right side of the pollution region $\mathcal{O}(h^s)$. That is, the pollution region size is almost of the same order $\mathcal{O}(h^{1/2})$ on both sides of the discontinuity, suggesting that our estimate about the pollution region size, $\mathcal{O}(h^{1/2} \log h^{-1})$, is sharp.

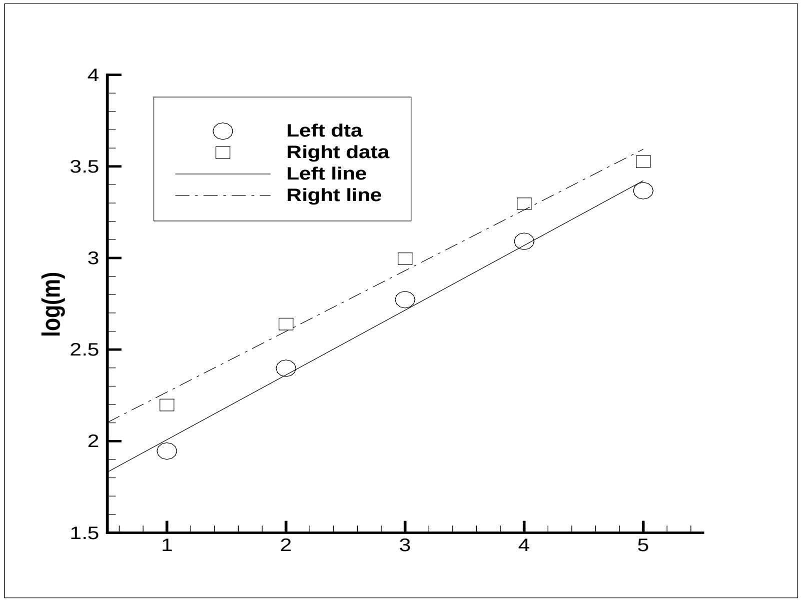


Figure 29: Dependence of the pollution region size and the mesh length. RKDG3 with $k = 2$.

If we repeat the process above for the $k = 3$ case (fourth order in space and third order in time, with $\Delta t = 0.18h^{4/3}$ so that the global error is fourth order), we obtain similar results. The least square process provides $s = 0.50000000000000097$ for the left boundary and $s = 0.499999999999999937$ for the right boundary of the pollution region $\mathcal{O}(h^s)$. That is, the pollution region size is almost of the same order $\mathcal{O}(h^{1/2})$ on both sides of the discontinuity, again suggesting that our estimate about the pollution region size, $\mathcal{O}(h^{1/2} \log h^{-1})$, is sharp for higher order k.

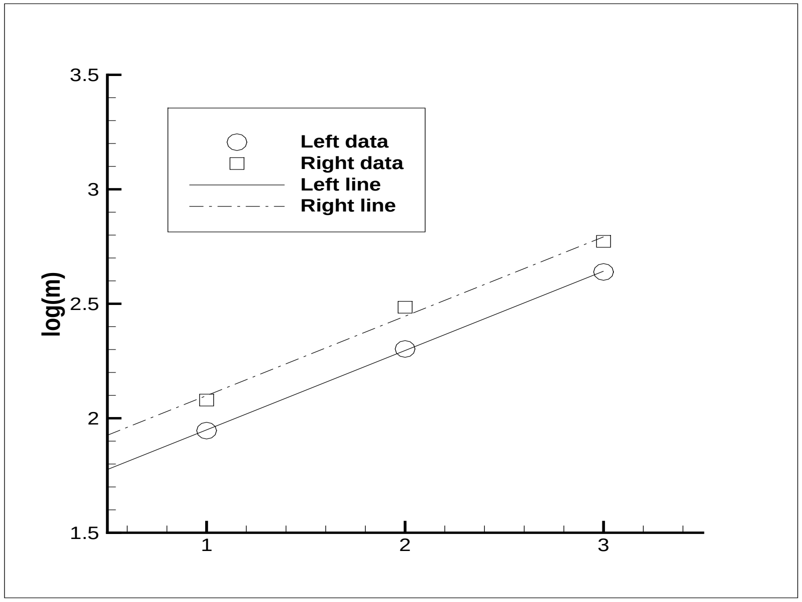


Figure 30: Dependence of the pollution region size and the mesh length. RKDG3 with $k = 3$.

- The result also holds for one-dimensional linear systems. The generalizations to two dimensions and especially to nonlinear hyperbolic equations are non-trivial and are left for future work.

DG method for hyperbolic equations with δ-singularities

We develop and analyze DG methods for solving hyperbolic conservation laws

$$\begin{aligned} &u_t + f(u)_x = g(x,t), && (x,t) \in R \times (0,T], \\ &u(x,0) = u_0(x), && x \in R, \end{aligned} \tag{17}$$

where the initial condition u_0, or the source term $g(x,t)$, or the solution $u(x,t)$ contains δ-singularities.

- Such problems appear often in applications and are difficult to approximate numerically, especially for finite difference schemes.
- Many numerical techniques rely on modifications with smooth kernels (mollification) and hence may severely smear such singularities, leading to large errors in the approximation.

In Yang and Shu, Num Math 13 and Yang, Wei and Shu, JCP 13, we develop, analyze and apply DG methods for solve hyperbolic equations with δ-singularities. The DG methods are based on weak formulations and can be designed directly to solve such problems without modifications, leading to very accurate results.

Linear equations with singular initial condition

We consider the linear model equation

$$\begin{aligned} u_t + \beta u_x &= 0 \\ u(x,0) = u^0(x) \end{aligned}$$

where β is a constant, $u_0(x)$ has compact support, has a sole δ-singularity at $x = 0$ and is sufficiently smooth everywhere else.

Even though the initial condition $u^0(x)$ is no longer in L^2, it does have an L^2-projection to the DG space V_h, which we use as the initial condition for the DG scheme. For problems involving δ-singularities, negative-order norm estimates are more natural. We have the following theorem in Yang and Shu, Num Math 13:

Theorem: By taking $\Omega_0 + 2supp(K_h^{2k+2,k+1}) \subset\subset \Omega_1 \subset\subset \Omega\backslash\mathcal{R}_T$, we have

$$\|u(T) - u_h(T)\|_{-(k+1)} \leq Ch^k, \tag{18}$$

$$\|u(T) - u_h(T)\|_{-(k+2)} \leq Ch^{k+1/2}, \tag{19}$$

$$\|u(T) - u_h(T)\|_{-(k+1),\Omega_1} \leq Ch^{2k+1}, \tag{20}$$

$$\|u(T) - K_h^{2k+2,k+1} * u_h(T)\|_{\Omega_0} \leq Ch^{2k+1}, \tag{21}$$

where the positive constant C does not depend on h. Here the mesh is assumed to be uniform for (21) but can be regular and non-uniform for the other three inequalities.

Several comments:

- We use the results about the pollution region in Zhang and Shu, Num Math to appear, which is also valid in the current case with δ-singularities.
- We follow the proof of negative-order error estimates and post-processing for DG methods solving linear hyperbolic equations with smooth solutions in Cockburn, Luskin, Shu and Süli Math Comp 2003 with suitable adjustments.

Numerical example: We solve the following problem

$$\begin{aligned} & u_t + u_x = 0, && (x,t) \in [0,\pi] \times (0,1], \\ & u(x,0) = \sin(2x) + \delta(x-0.5), && x \in [0,\pi], \end{aligned} \tag{22}$$

with periodic boundary condition $u(0,t) = u(\pi,t)$. Clearly, the exact solution is

$$u(x,t) = \sin(2x-2t) + \delta(x-t-0.5).$$

Table 4: L^2-norm of the error between the numerical solution and the exact solution for equation (22) after post-processing in the region away from the singularity.

		$\mathcal{P}^1$ polynomial		$\mathcal{P}^2$ polynomial		$\mathcal{P}^3$ polynomial	
N	d	error	order	error	order	error	order
200	0.2	6.88E-05	-	8.40e-07	-	1.48E-09	-
300	0.2	1.41E-05	3.92	3.56e-10	19.2	3.98E-13	20.3
400	0.2	5.89E-06	3.02	1.98e-11	10.1	4.42E-16	23.7
500	0.2	3.01E-06	3.01	6.13e-12	5.25	7.49E-17	7.95
600	0.2	1.74E-06	3.00	2.37e-12	5.21	1.76E-17	7.94

We consider the following two dimensional problem

$$\begin{aligned} &u_t + u_x + u_y = 0, && (x,y,t) \in [0,2\pi] \times [0,2\pi] \\ &u(x,0) = \sin(x+y) + \delta(x+y-2\pi), && (x,y) \in [0,2\pi] \times [0,2\pi], \end{aligned} \tag{23}$$

with periodic boundary condition. Clearly, the exact solution is

$$u(x,t) = \sin(x+y-2t) + \delta(x+y-2t) + \delta(x+y-2t-2\pi).$$

We use $\mathcal{Q}^k$ polynomial approximation spaces with $k=1$ and 2.

Table 5: L^2-norm of the error between the numerical solution and the exact solution for equation (23) after post-processing in the region away from the singularity.

		$\mathcal{Q}^1$ polynomial		$\mathcal{Q}^2$ polynomial	
N	d	error	order	error	order
400	0.4	2.60E-05	-	3.23e-08	-
500	0.4	1.24E-05	3.32	2.47e-10	20.0
600	0.4	7.16E-06	3.01	1.19e-11	16.6
700	0.4	4.50E-06	3.01	5.11e-12	5.47
800	0.4	3.01E-06	3.02	2.53e-12	5.29

The theory generalizes to linear systems in a straightforward way. We solve the following linear system

$$\begin{aligned}
&u_t - v_x = 0, && (x,t) \in [0,2] \times (0,0.4], \\
&v_t - u_x = 0, && (x,t) \in [0,2] \times (0,0.4], \\
&u(x,0) = \delta(x-1), v(x,0) = 0, && x \in [0,2].
\end{aligned} \tag{24}$$

Clearly, the exact solution (the Green's function) is

$$u(x,t) = \frac{1}{2}\delta(x-1-t) + \frac{1}{2}\delta(x-1+t),$$

$$v(x,t) = \frac{1}{2}\delta(x-1+t) - \frac{1}{2}\delta(x-1-t).$$

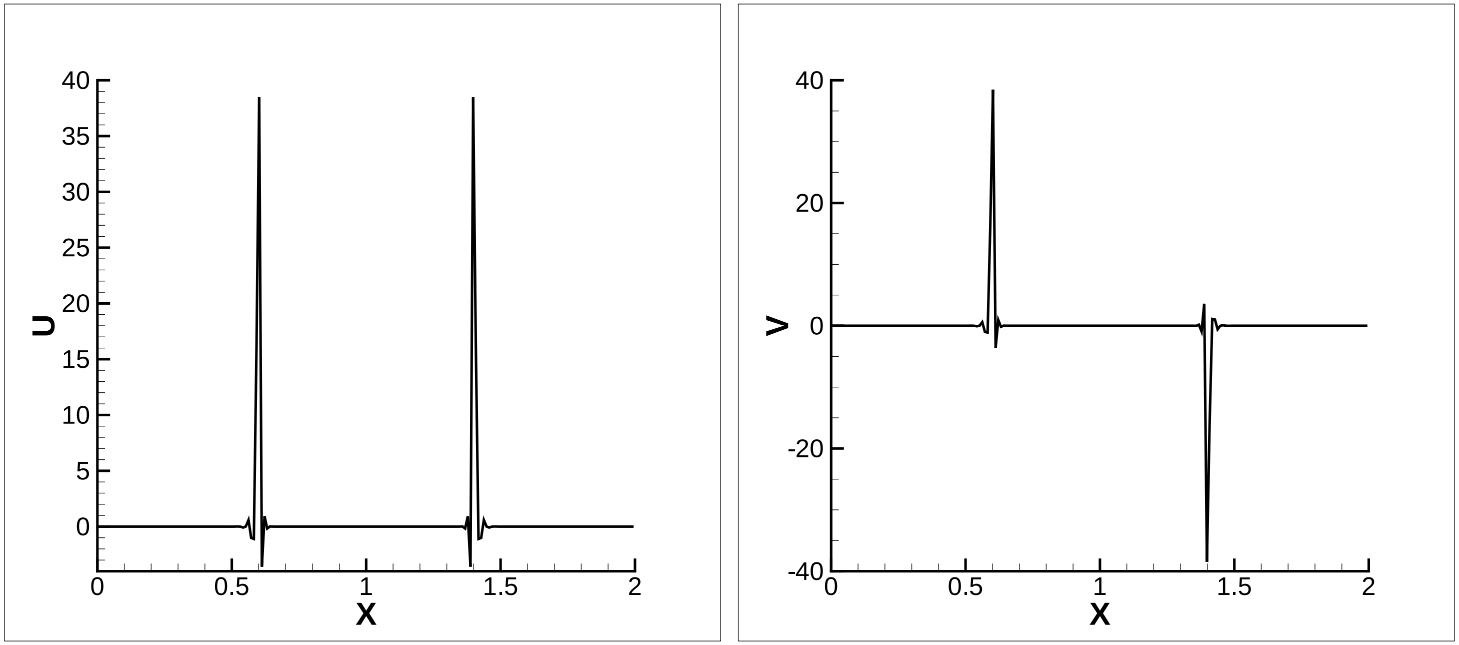


Figure 31: Solutions of u (left) and v (right) for (24) at $t = 0.4$.

Linear equations with singular source terms

We consider the linear model equation

$$\begin{aligned} u_t(x,t) + Lu(x,t) &= g(x,t), \qquad (x,t) \in \Omega \times (0,\infty), \\ u(x,0) &= 0, \qquad x \in \Omega, \end{aligned}$$

with L being a linear differential operator that does not involve time derivatives and $g(x,t)$ is a singular source term, for example $g(x,t) = \delta(x)$. The singular source term can be implemented in the DG scheme in a straightforward way, since it involves only the integrals of the singular source term with test functions in V_h.

By using Duhamel's principle, we can prove the following theorem (Yang and Shu, Num Math 13):

Theorem: Denote $\mathcal{R}_T = I_i \cup (T - C\log(1/h)h^{1/2}, T + C\log(1/h)h^{1/2})$, where I_i is the cell which contains the concentration of the δ-singularity on the source term. Then we have the following estimates

$$\|u(T) - u_h(T)\|_{-(k+1)} \leq Ch^k, \tag{25}$$

$$\|u(T) - u_h(T)\|_{-(k+2)} \leq Ch^{k+1/2}, \tag{26}$$

$$\|u - u_h\|_{-(k+1),\Omega_1} \leq Ch^{2k+1}, \tag{27}$$

$$\|u(T) - K_h^{2k+2,k+1} * u_h(T)\|_{\Omega_0} \leq Ch^{2k+1}, \tag{28}$$

where $\Omega_0 + 2supp(K_h^{2k+2,k+1}) \subset\subset \Omega_1 \subset\subset \mathbb{R}\backslash\mathcal{R}_T$. Here the mesh is assumed to be uniform for (28) but can be regular and non-uniform for the other three inequalities.

Numerical example: We solve the following problem

$$\begin{aligned} & u_t + u_x = \delta(x-\pi), && (x,t) \in [0,2\pi] \times (0,1], \\ & u(x,0) = \sin(x), && x \in [0,2\pi], \\ & u(0,t) = 0, && t \in (0,1]. \end{aligned} \tag{29}$$

Clearly, the exact solution is

$$u(x,t) = \sin(x-t) + \chi_{[\pi,\pi+t]},$$

where $\chi_{[a,b]}$ denotes the indicator function of the interval $[a,b]$.

Table 6: L^2-norm of the error between the numerical solution and the exact solution for equation (29) after post-processing in the region away from the singularity.

		$\mathcal{P}^1$ polynomial		$\mathcal{P}^2$ polynomial	
N	d	error	order	error	order
401	0.2	1.74E-06	-	4.29E-08	-
801	0.2	5.92E-09	8.22	6.80E-13	15.9
1601	0.2	7.36E-10	3.03	1.34E-17	12.3
3201	0.2	9.19E-11	3.01	3.86E-18	5.13
6401	0.2	1.15E-11	3.01	1.16E-19	5.07

Rendez-vous algorithm

Even though our theory is established only for linear equations, the DG algorithm can be easily implemented for nonlinear hyperbolic equations involving δ-singularities.

In Canuto, Fagnani and Tilli, SIAM J Control and Optimization 2012, the following problem

$$\begin{aligned} &\rho_t + F_x = 0, \qquad && x \in [0,1], t > 0, \\ &\rho(0,t) = u_0(x), && t > 0, \end{aligned} \tag{30}$$

is studied. Here ρ is the density function, which is always positive.

The flux F is given by

$$F(t,x) = v(t,x)\rho(t,x),$$

and the velocity v is defined by

$$v(t,x) = \int_{\mathbb{R}^n} (y-x)\xi(y-x)\rho(t,y)dy,$$

where $\xi(x)$ is a positive function and supported on a ball centered at zero with radius R. Canuto et al. proved that when t tends to infinity, the density function ρ will converge to some δ-singularities, and the distances between any of them cannot be less than R. Some computational results are shown in Canuto et al. based on a first order finite volume method.

We use the DG algorithm with the positivity-preserving limiter in Zhang and Shu JCP 2010, which can maintain positivity without affecting the high order accuracy, to both the one and two dimensional Rendez-vous algorithms, in Yang and Shu, Num Math 13 and Yang, Wei and Shu, JCP 13.

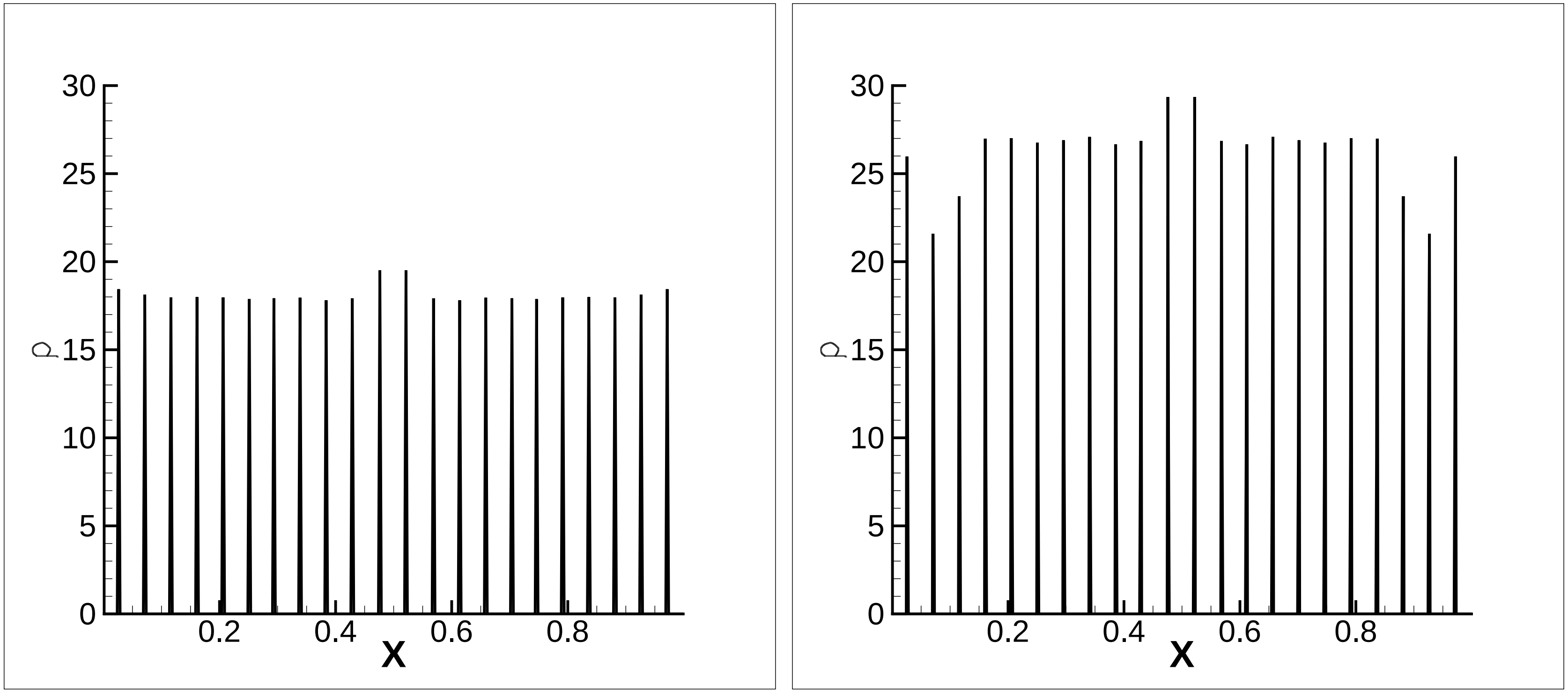


Figure 32: Numerical density at $t = 1000$ with $N = 400$ when using $\mathcal{P}^0$ (left) and $\mathcal{P}^1$ (right) polynomials.

In 2D, the model is

$$\begin{aligned} &\rho_t + \text{div}(\mathbf{v}\rho) = 0, && \mathbf{x} \in [-1,1]^2, t > 0, \\ &\rho(\mathbf{x},0) = \rho_0(\mathbf{x}), && t > 0, \end{aligned} \tag{31}$$

where the velocity $\mathbf{v}$ is defined by

$$\mathbf{v}(\mathbf{x},t) = \int_{B_R(\mathbf{x})} (\mathbf{y}-\mathbf{x})\rho(\mathbf{y},t)d\mathbf{y}.$$

In this example, we take $R = 0.1$ and

$$\rho_0(\mathbf{x}) = \begin{cases} 1 & r < 0.5, \\ 0 & r > 0.5, \end{cases}$$

where $r = \|\mathbf{x}\|$ is the Euclidean norm of $\mathbf{x}$.

In Canuto et al., the authors demonstrated that the exact solution should be a single delta placed at the origin.

However, when we use rectangle meshes, we observe more than one delta singularity for R sufficiently small. This is because the meshes are not invariant under rotation.

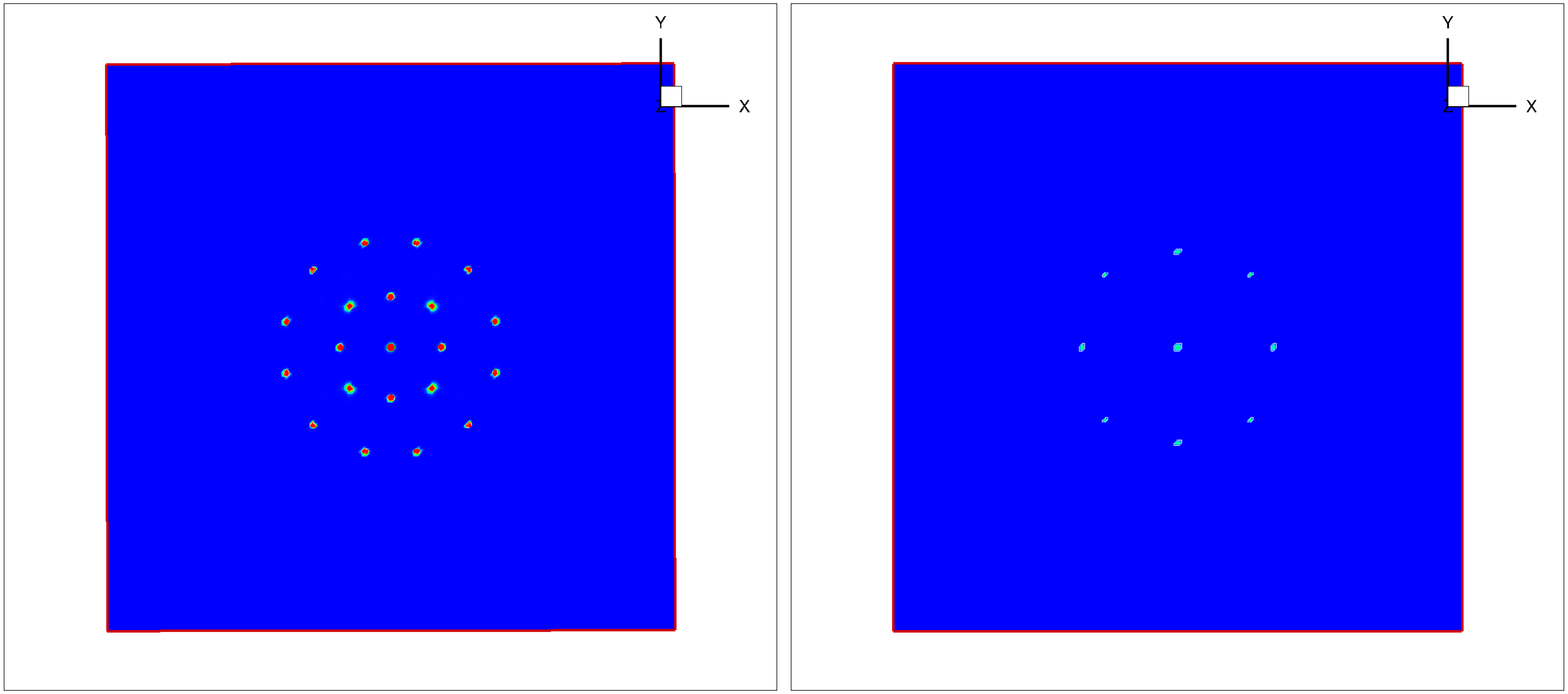


Figure 33: Numerical density ρ with a rectangular 100×100 mesh using $\mathcal{P}^0$ elements. $R = 0.08$ (left) and $R = 0.1$ (right).

To tackle this problem, we follow the same ideas in Cheng and Shu, JCP 2010; CiCP 2012, and construct a special equal-angle-zoned mesh. The structure of the mesh is given in figure 34. By using such a special mesh, the limit density is a single delta placed at the origin.

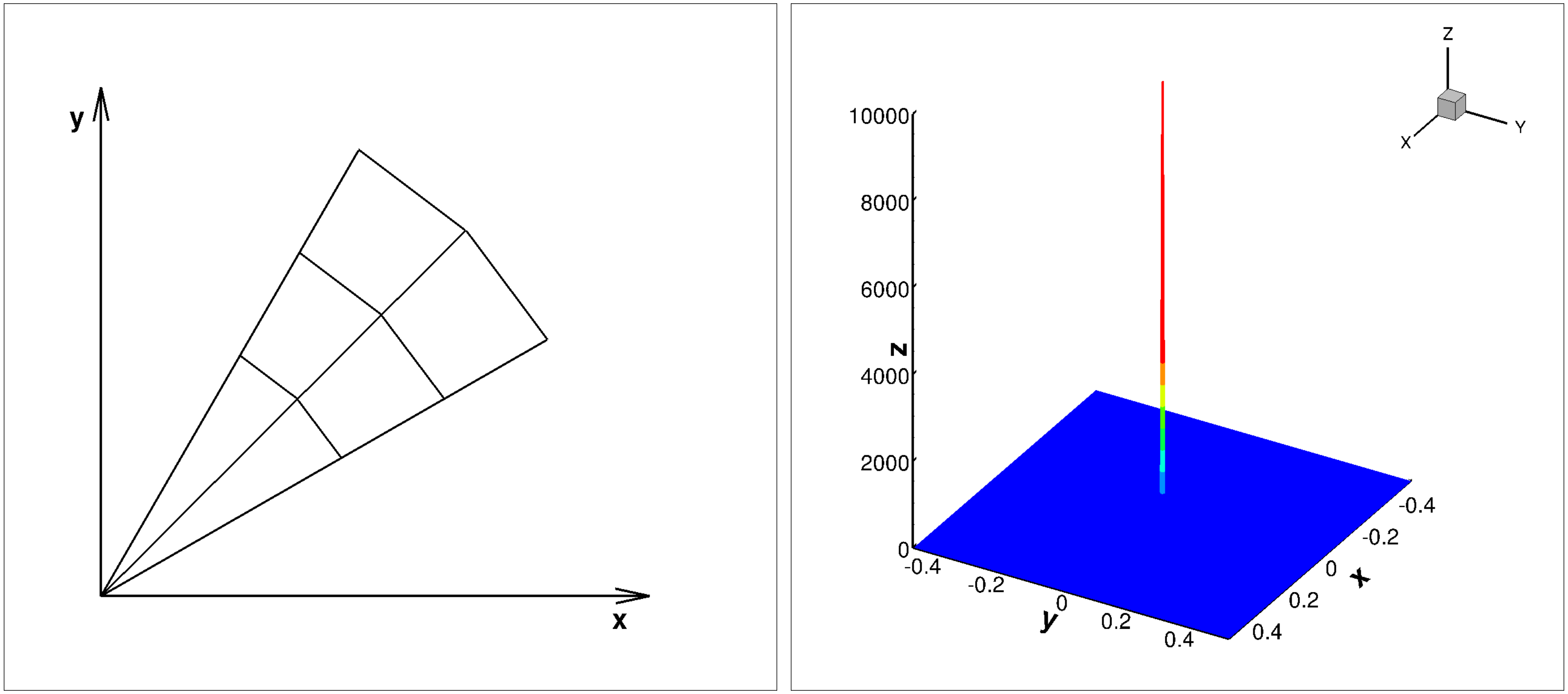


Figure 34: Left: Equal-angle-zoned mesh. Right: Numerical density ρ for (31) at $t = 2000$ with $N = 200$ using $\mathcal{P}^0$ elements.

Pressureless Euler equations

Another important system admitting δ-singularities in its solutions is the pressureless Euler equation

$$\mathbf{w}_t + \mathbf{f}(\mathbf{w})_x = 0, \quad t > 0,\ x \in \mathbb{R}, \tag{32}$$

$$\mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix}, \quad \mathbf{f}(\mathbf{w}) = \begin{pmatrix} m \\ \rho u^2 \end{pmatrix},$$

with $m = \rho u$, where ρ is the density function and u is the velocity.

It is quite difficult to obtain stable schemes for solve this system, especially for high order schemes.

A good property of this system is that the density is always positive and the velocity satisfies a maximum-principle. Thus, in 1D, the convex set

$$G = \left\{ \mathbf{w} = \begin{pmatrix} \rho \\ m \end{pmatrix} : \rho > 0, a\rho \leq m \leq b\rho \right\},$$

where

$$a = \min u_0(x), \qquad b = \max u_0(x), \tag{33}$$

with u_0 being the initial velocity, is invariant. In Yang, Wei and Shu, JCP 13, we adapt the techniques in Zhang and Shu, JCP 2010 to design a limiter to guarantee that our DG solution stays in set G without affecting high order accuracy. This is also generalized to 2D. Our scheme is thus very robust, stable and high order accurate for this pressureless Euler system.

We consider the following initial data

$$\rho_0(x) = \sin(x) + 2,\ u_0(x) = \sin(x) + 2, \tag{34}$$

with periodic boundary condition. Clearly, the exact solution is

$$u(x,t) = u_0(x_0), \quad \rho(x,t) = \frac{\rho_0(x_0)}{1 + u_0'(x_0)},$$

where x_0 is given implicitly by

$$x_0 + t u_0(x_0) = x.$$

Table 7: L^2-norm of the error between the numerical density and the exact density for initial condition (34).

	k=1		k=2		k=3	
N	error	order	error	order	error	order
20	1.41E-02	-	6.84E-04	-	3.40e-5	-
40	4.18E-03	1.76	1.04E-04	2.72	2.82e-6	3.59
80	1.30E-03	1.68	1.55E-05	2.74	2.26e-7	3.64
160	4.24E-04	1.62	2.41E-06	2.69	1.83e-8	3.62
320	1.51E-04	1.49	3.80E-07	2.67	1.49e-9	3.63

We consider the following initial condition

$$\rho_0(x) = \begin{cases} 1 & x < 0, \\ 0.25 & x > 0, \end{cases} \qquad u_0(x) = \begin{cases} 1 & x < 0, \\ 0 & x > 0. \end{cases} \tag{35}$$

Clearly, the exact solution is

$$(\rho(x,t), u(x,t)) = \begin{cases} (1,1) & x < 2t/3, \\ (0.25, 0) & x > 2t/3, \end{cases}$$

and at $x = \frac{2t}{3}$, the density should be a δ-function.

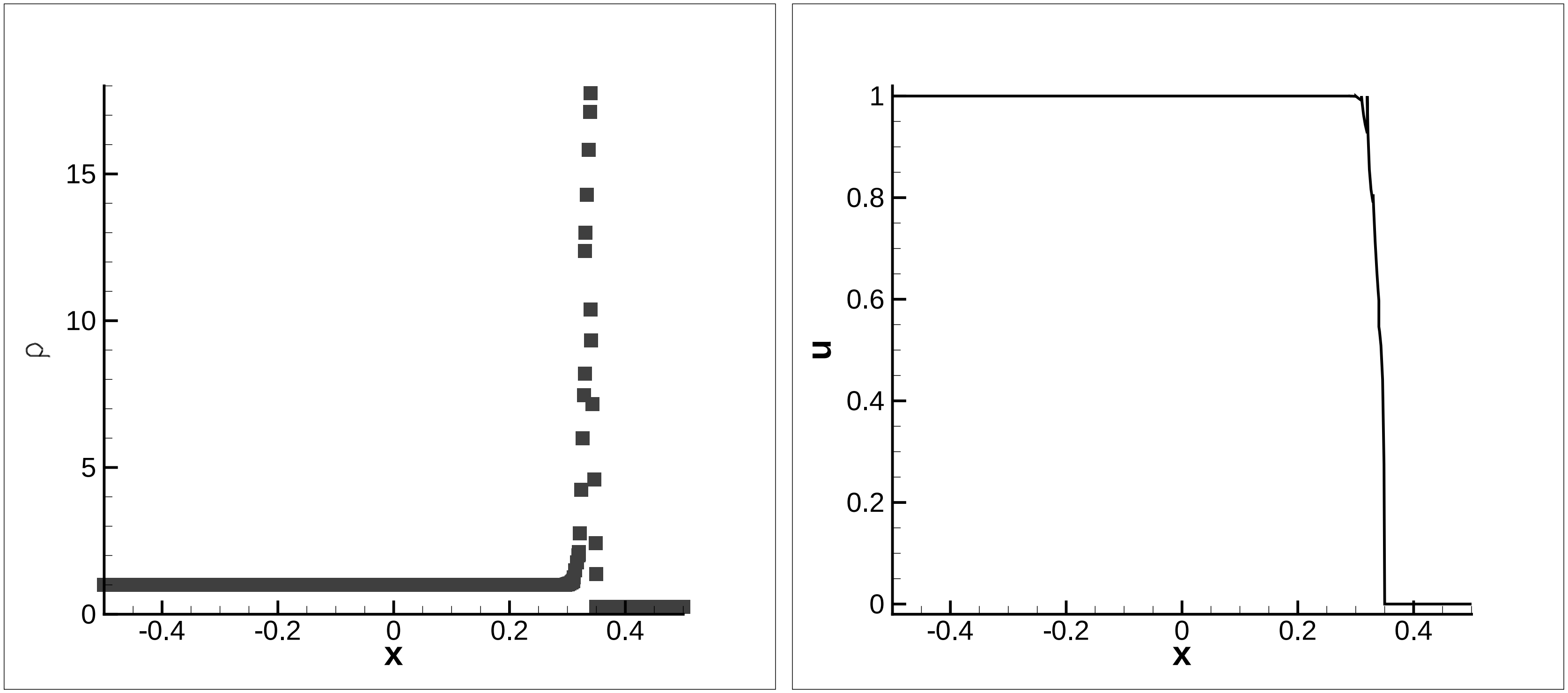


Figure 35: Numerical density (left) and velocity (right) at $t = 0.5$ with $\mathcal{P}^1$ polynomials for initial condition (35).

We consider the following initial condition

$$\begin{aligned}\rho(x,y,0) &= \rho_0(x+y) = \exp(\sin(x+y)),\\ u(x,y,0) &= u_0(x+y) = \tfrac{1}{3}(\cos(x+y)+2),\\ v(x,y,0) &= v_0(x+y) = \tfrac{1}{3}(\sin(x+y)+2).\end{aligned} \tag{36}$$

The exact solution is

$$u(x,y,t) = u_0(z_0), \quad v(x,y,t) = v_0(z_0), \quad \rho(x,y,t) = \frac{\rho_0(z_0)}{1+u_0'(z_0)+v_0'(z}$$

where z_0 is given implicitly by

$$z_0 + t(u_0(z_0)+v_0(z_0)) = x+y.$$

Table 8: L^2-norm of the error between the numerical density and the exact density for initial condition (36).

	k=1		k=2		k=3	
N	error	order	error	order	error	order
10	0.512	-	0.107	-	3.42E-02	-
20	0.176	1.54	3.12E-02	1.78	3.57E-03	3.26
40	6.48E-02	1.44	8.52E-03	1.87	4.86E-04	2.88
80	2.32E-02	1.48	1.39E-03	2.62	3.97E-05	3.61
160	9.08E-03	1.35	1.92E-04	2.86	3.65E-06	3.45

We consider the following initial condition

$$\rho(x,y,0) = \frac{1}{100}, \quad (u,v)(x,y,0) = (-\frac{1}{10}\cos\theta, -\frac{1}{10}\sin\theta), \tag{37}$$

where θ is the polar angle.

Since all the particles are moving towards the origin, the density function at $t > 0$ should be a single delta at the origin.

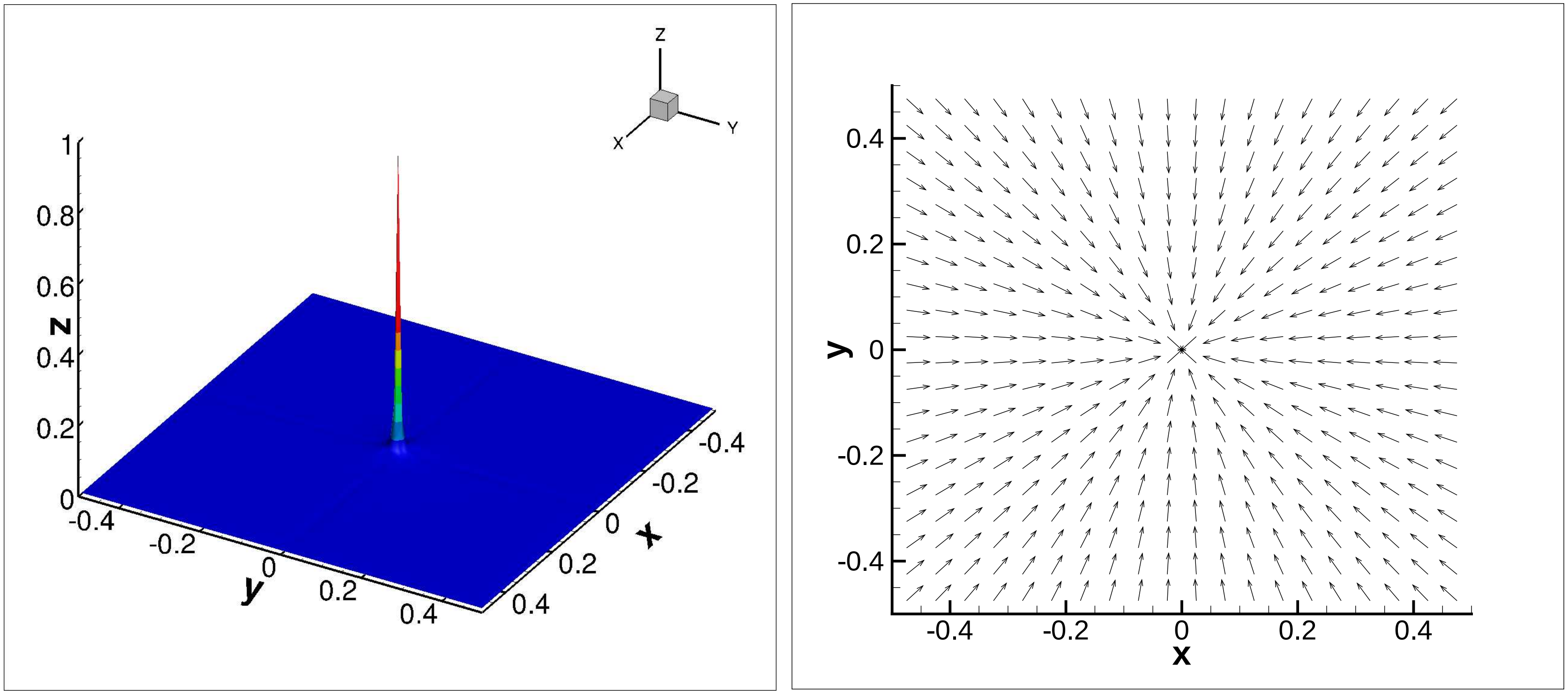


Figure 36: Numerical density (left) and velocity field (right) at $t = 0.5$ for the initial condition (37).

We consider the following initial condition

$$\rho(x,y,0) = \frac{1}{10}, \quad (u,v)(x,y,0) = \begin{cases} (-0.25,-0.25) & x>0, y>0, \\ (0.25,-0.25) & x<0, y>0, \\ (0.25,0.25) & x<0, y<0, \\ (-0.25,0.25) & x>0, y<0. \end{cases} \tag{38}$$

Figure 37 shows the numerical density and velocity field at $t = 0.5$. From the figure, we can observe δ-singularities located at the origin and the two axes.

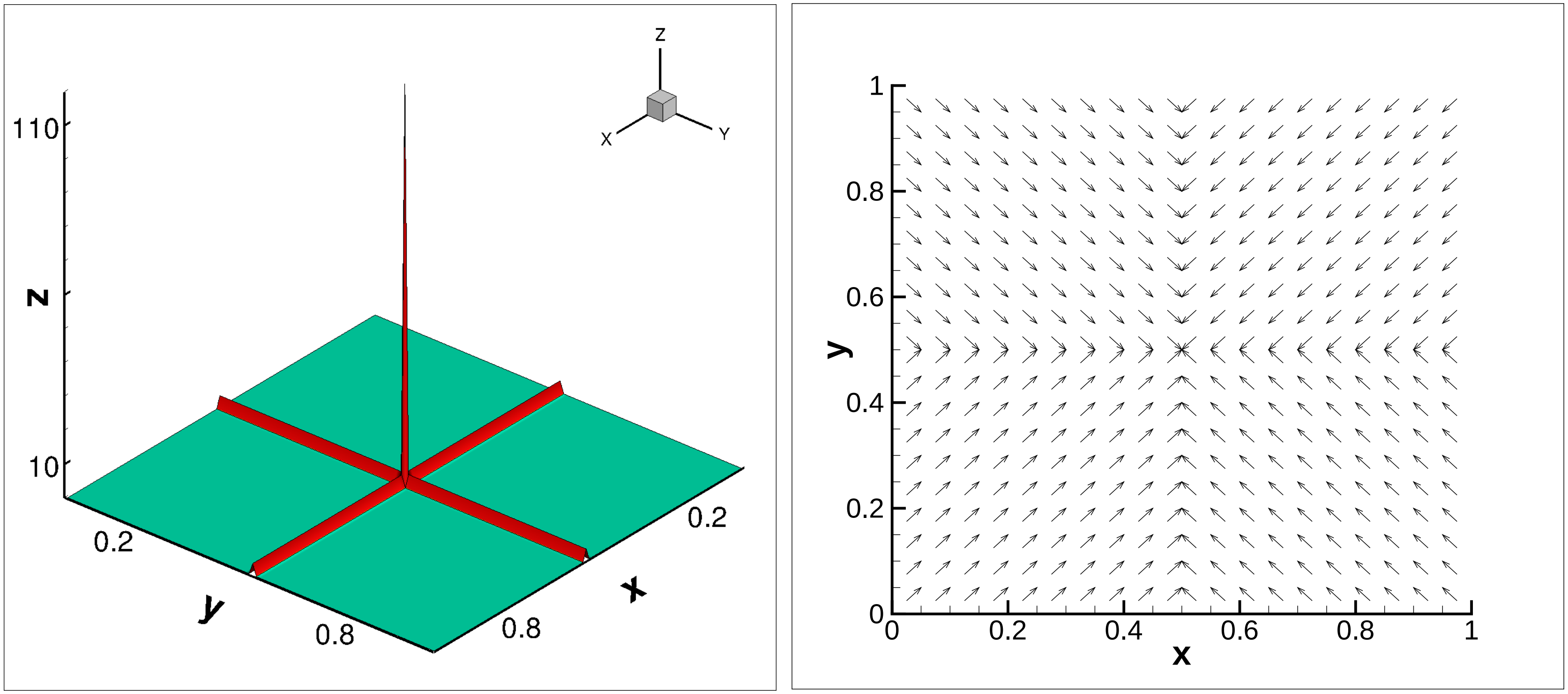


Figure 37: Numerical density (left) and velocity field (right) at $t = 0.5$ for initial condition (38).

We consider the following initial condition

$$\rho(x,y,0)=\frac{1}{100},\quad (u,v)(x,y,0)=\begin{cases}(\cos\theta,\sin\theta) & r<0.3,\\ (-\frac{1}{2}\cos\theta,-\frac{1}{2}\sin\theta) & r>0.3,\end{cases} \tag{39}$$

where $r=\sqrt{x^2+y^2}$ and θ is the polar angle.

Figure 38 shows the numerical density (contour plot) and velocity field at $t=0.5$. From the figure, we can observe δ-shocks located on a circle and vacuum inside.

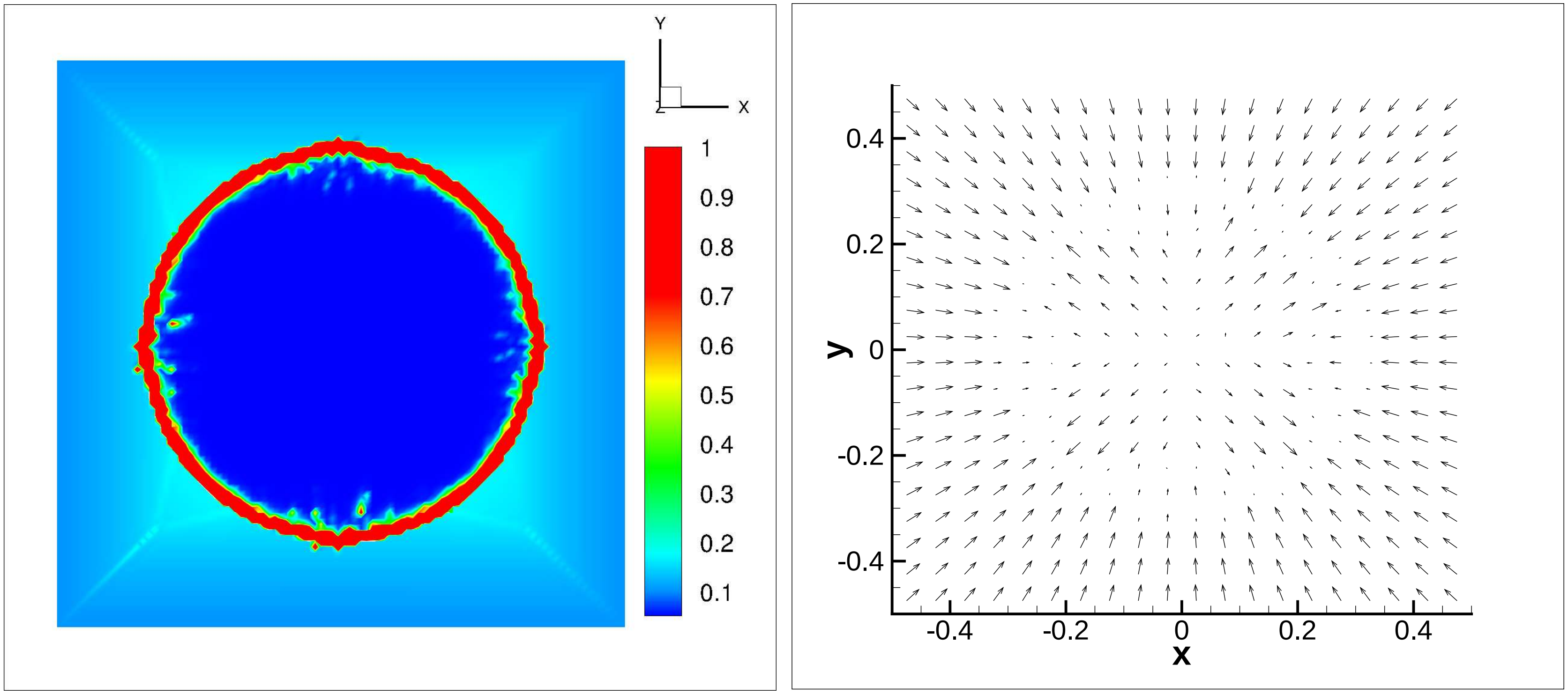


Figure 38: Numerical density (left) and velocity field (right) at $t = 0.5$ for initial condition (39).

We consider the following initial condition

$$\rho(x,y,0) = 0.5, \quad (u,v)(x,y,0) = \begin{cases} (0.3, 0.4) & x > 0, y > 0, \\ (-0.4, 0.3) & x < 0, y > 0, \\ (-0.3, -0.4) & x < 0, y < 0, \\ (0.4, -0.3) & x > 0, y < 0. \end{cases} \tag{40}$$

Figure 39 shows the numerical density (contour plot) and velocity field with $N = 50$ at $t = 0.4$. From the figure, we can observe that the numerical solution approximates the vacuum quite well.

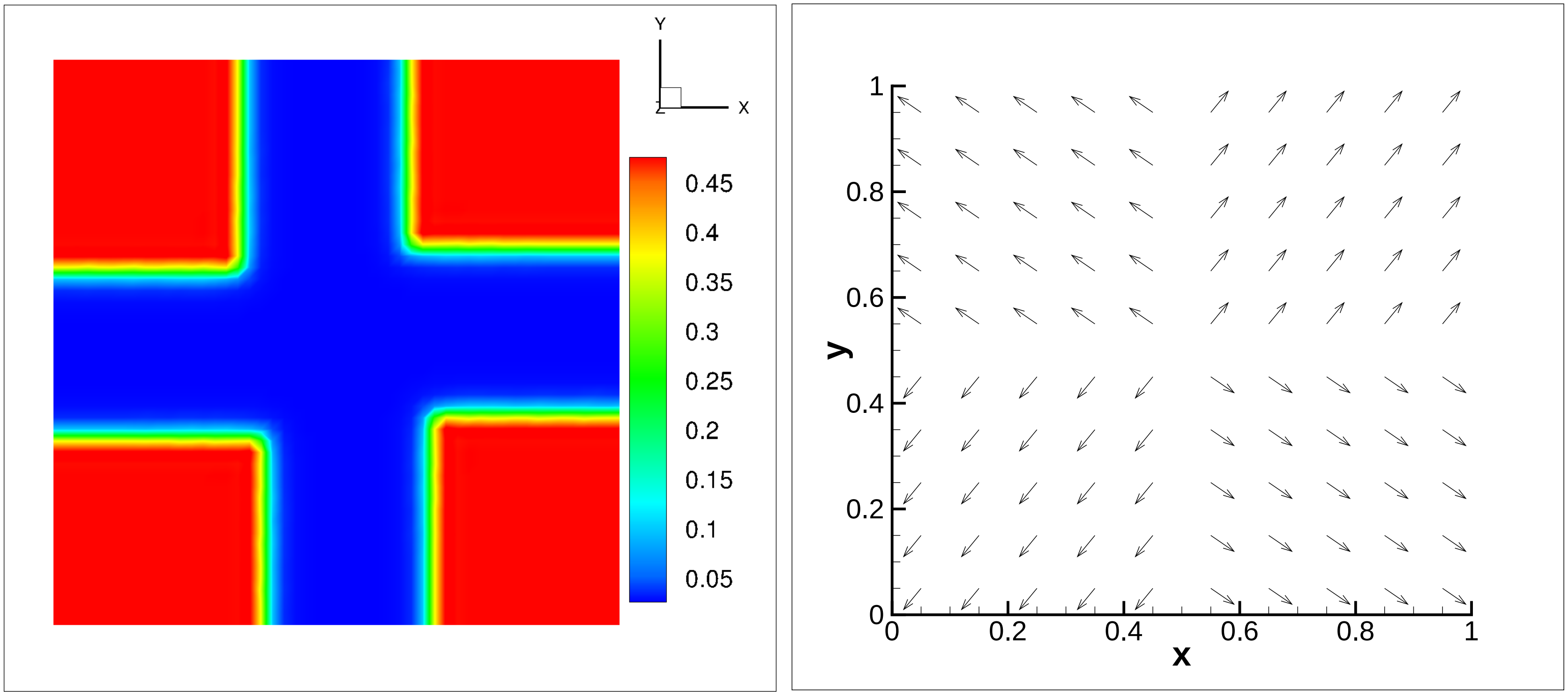


Figure 39: Numerical density (left) and velocity field (right) at $t = 0.4$ with $N = 50$ for initial condition (40).

Concluding remarks for DG with singularities

- DG methods are suitable for computing solutions with discontinuities or δ-singularities, because it satisfies a cell entropy inequality and is based on a weak formulation.

- For linear model equations, the DG methods can be shown to converge in optimal L^2 error $\mathcal{O}(h^{1/2}\log(1/h))$ away from the singularity for discontinuous but piecewise smooth solutions.

- For linear model equations, the DG methods can be shown to converge in negative norms when either the initial condition or the source term contains δ-singularities. This convergence is of $\mathcal{O}(h^{k+1/2})$ order in the whole domain and of $\mathcal{O}(h^{2k+1})$ order $\mathcal{O}(h^{1/2}\log(1/h))$ away from the singularities. Post-processing then produces $\mathcal{O}(h^{2k+1})$ order superconvergence in the strong L^2 norm $\mathcal{O}(h^{1/2}\log(1/h))$ away from the singularities.

- DG methods work well for nonlinear problems containing δ-singularities. It is usually important to design and apply a bound-preserving limiter which keeps high order accuracy and can effectively prevent nonlinear instability caused by overshoots of the numerical solution into the ill-posed regime of the nonlinear PDEs.

The End

THANK YOU!