

An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary

Jianfang Lu ^{*}, Chi-Wang Shu [†], Sirui Tan [‡], and Mengping Zhang [§]

October 18, 2020

Abstract

In this paper, we reconsider the inverse Lax-Wendroff (ILW) procedure, which is a numerical boundary treatment for solving hyperbolic conservation laws, and propose a new approach to evaluate the values on the ghost points. The ILW procedure was firstly proposed to deal with the “cut cell” problems, when the physical boundary intersects with the Cartesian mesh in an arbitrary fashion. The key idea of the ILW procedure is repeatedly utilizing the partial differential equations (PDEs) and inflow boundary conditions to obtain the normal derivatives of each order on the boundary. A simplified ILW procedure was proposed in [28] and used the ILW procedure for the evaluation of the first order normal derivatives only. The main difference between the simplified ILW procedure and the proposed ILW procedure here is that we define the unknown u and the flux $f(u)$ on the ghost points separately. One advantage of this treatment is that it allows the eigenvalues of the Jacobian $f'(u)$ to be close to zero on the boundary, which may appear in many physical problems. We also propose a new weighted essentially non-oscillatory (WENO) type extrapolation at

^{*}South China Research Center for Applied Mathematics and Interdisciplinary Studies, South China Normal University, Canton, Guangdong 510631, China. E-mail: jflu@m.scnu.edu.cn. J. Lu’s research is partially supported by NSFC grant 11901213.

[†]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: chi-wang_shu@brown.edu. C.-W. Shu’s research is partially supported by NSF grants DMS-1719410 and DMS-2010107, and AFOSR grant FA9550-20-1-0055.

[‡]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: sirui_tan@alumni.brown.edu.

[§]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: mpzhang@ustc.edu.cn. M. Zhang’s research is partially supported by NSFC grant 11871448.

the outflow boundaries, whose idea comes from the multi-resolution WENO schemes in [32]. The WENO type extrapolation maintains high order accuracy if the solution is smooth near the boundary and it becomes a low order extrapolation automatically if a shock is close to the boundary. This WENO type extrapolation preserves the property of self-similarity, thus it is more preferable in computing the hyperbolic conservation laws. We provide extensive numerical examples to demonstrate that our method is stable, high order accurate and has good performance for various problems with different kinds of boundary conditions including the solid wall boundary condition, when the physical boundary is not aligned with the grids.

Key Words: Hyperbolic conservation laws; Inverse Lax-Wendroff method; Numerical boundary condition; WENO type extrapolation; Solid wall

1 Introduction

In this paper, we consider a numerical boundary treatment for solving hyperbolic conservation laws with high order finite difference methods on the Cartesian mesh. The Cartesian mesh is attractive and preferable for its simple structure and easy generation, and it allows the use of the high-resolution shock capturing methods that are more complicated to develop on unstructured meshes. As mentioned in [25], there are two kinds of difficulties that should be treated carefully when imposing the inflow boundary conditions with the high order finite difference schemes. One is the treatment of the ghost points near the boundary because of the wide stencils of the interior scheme. Another difficulty is that the mesh may not be aligned with the boundaries of the geometric body, then the so-called “cut cell” problem arises. This problem would cause some numerical difficulties. For instance, in finite volume methods it may lead to a restricted time step condition, and the h -box method was proposed in [3] to overcome this difficulty. There are many attempts to deal with the “cut cell” problems, such as the embedded boundary method [11, 22, 14], immersed boundary method [17, 15], and the references therein. In this paper, we focus on the inverse Lax-Wendroff (ILW) procedure, which was first proposed by Tan and Shu in [25] to deal with the inflow boundary conditions when solving hyperbolic conservation laws. The idea of the ILW procedure comes from the Lax-Wendroff type boundary condition procedure [9, 31], in which the authors repeatedly used the PDEs to write the normal derivatives to the inflow boundary in terms of the tangential derivatives, for solving the static Hamilton-Jacobi equation with high order fast sweeping WENO methods. Tan and Shu extended this procedure to solve the time-dependent hyperbolic conservation laws in static or moving geometries in [25, 26]. The essence of the ILW method is repeatedly utilizing the partial differential equations (PDEs)

to obtain the normal spatial derivatives on the inflow boundary, in terms of the time and tangential derivatives of the given boundary condition. For earlier related work of this procedure, see [7, 8]. The core idea of computing the higher order derivatives of the solution via the governing equation dates back to the classical scheme of Lax and Wendroff, that is why the current method is called ILW. This idea has been used extensively over the past years for building discretizations of both differential equations and boundary conditions. For example, in [23], Singer and Turkel introduced the equation-based compact finite difference schemes for the Helmholtz equation. In the approach of [23], the derivatives of the solution are needed to cancel the leading terms in the expansion of the truncation error and thus achieve high order accuracy on a small (i.e., compact) stencil. Subsequently, similar ideas have been explored and developed further, e.g. by Baruch, Fibich, and Tsynkov [2] and Turkel et al. [24]. Moreover, the method of difference potentials by Ryaben’kii [18] also uses the equation-based differentiation to enable the treatment of non-conforming boundaries on simple grids (such as Cartesian) with no loss of accuracy. It has been applied to a variety of 2D and 3D wave propagation problems in both frequency domain [16] and time domain [4].

Due to the heavy algebra of the ILW procedure for 2D nonlinear systems, Tan et al. developed a simplified and improved implementation of this procedure for hyperbolic systems with source terms in [28]. The stability analysis of the ILW procedure can be found in [25, 29, 13]. Also, this procedure has been extended to other problems such as Boltzmann type models [5], convection-diffusion problems [12], etc. For the survey and developments of the ILW procedure, see [27, 20].

In this paper, we study the ILW methods developed in [25, 26], and propose a new ILW method to solve hyperbolic conservation laws. In [25, 26], it is required that the eigenvalues of the Jacobian $f'(u)$ cannot be too close to zero on the boundary. We aim at removing this restriction by evaluating the unknown u and the flux $f(u)$ independently, thus keeping the eigenvalues away from being the denominator. Same as in [26], we only perform the ILW procedure for the evaluation of the first order normal derivative, while all higher order derivatives are obtained by extrapolation. At the outflow boundary, Tan and Shu in [25, 26] used the classical Lagrangian extrapolation or least squares extrapolation when the solution is smooth, or the WENO type extrapolation if there is a shock near the boundary. However, the weights of the WENO type extrapolation in [25, 26] depend explicitly on the mesh size, hence violating the self-similarity property of the finite difference WENO schemes. In this paper, we adopt the idea of the multi-resolution WENO method in [32], and propose a new WENO type extrapolation. The linear weights in the multi-resolution WENO procedure can be arbitrary, and we choose them as some suitable positive numbers in the new WENO

type extrapolation. This procedure works well in all the numerical examples. We remark that our treatment also works well for the problems with the solid wall boundary condition.

This paper is organized as follows. In Section 2, we propose our numerical boundary treatment for both one-dimensional and two-dimensional hyperbolic conservation laws with the fifth order finite difference WENO method as an example. For the one-dimensional hyperbolic conservation laws, we first briefly review the original ILW method at the inflow boundary and the WENO type extrapolation at the outflow boundary in [25], then we introduce the new ILW method and the new WENO type extrapolation, and extend them to the two-dimensional problems. In Section 3, we provide a variety of numerical examples on accuracy tests and some benchmark problems, to demonstrate the effectiveness and robustness of the proposed algorithm. Concluding remarks are given in Section 4 .

2 Scheme formulation

In this section, we present an inverse Lax-Wendroff (ILW) procedure for treating the boundary conditions. As we shall see later, this new treatment will prevent the eigenvalues of the Jacobian $f'(u)$ from appearing in the denominators, thus it can be applied for the cases when the eigenvalues are close to zero. We also consider another kind of WENO type extrapolation, which preserves the property of self-similarity. We begin with the one-dimensional conservation laws to illustrate our idea, and then extend the algorithm to two-dimensional systems.

2.1 One-dimensional scalar conservation laws

First let us briefly review the ILW method for hyperbolic equations [25], to explain the basic idea and set notations. For simplicity, we consider the following one-dimensional equation

$$\begin{cases} u_t + f(u)_x = 0, & a < x < b, t > 0, \\ u(x, 0) = u_0(x), & a < x < b, \\ u(a, t) = g(t), & t > 0. \end{cases} \quad (2.1)$$

Without loss of generality, we assume $f'(u) > \alpha > 0$ at $x = a$ and $x = b$, with α being a positive constant. Therefore, we have an inflow boundary condition at $x = a$ and an outflow boundary condition at $x = b$. For two constants $\delta_1, \delta_2 \in [0, 1)$, we use a uniform mesh with the mesh size $\Delta x = (b - a)/(N + \delta_1 + \delta_2)$ and distribute the grid points as

$$x_j = a + (j + \delta_1) \Delta x, \quad j = -3, \dots, N + 3. \quad (2.2)$$

then $\{x_0, x_1, \dots, x_N\}$ are our interior points and the closest interior points to the left and right boundaries are $x_0 = a + \delta_1 \Delta x$ and $x_N = b - \delta_2 \Delta x$ respectively. Notice that we have deliberately allowed the physical boundaries $x = a$ and $x = b$ not located on the grid points.

In this paper, we use the fifth order upwind-biased conservative finite difference operator with the Lax-Friedrichs flux splitting technique (see e.g. [10]) to approximate the first order spatial derivative. The semi-discrete scheme for (2.1) is given as

$$(u_j)_t + \frac{1}{\Delta x} (\hat{f}_{j+\frac{1}{2}} - \hat{f}_{j-\frac{1}{2}}) = 0, \quad j = 0, 1, \dots, N, \quad (2.3)$$

where u_j is the approximation of u at $x = x_j$. For the fifth order WENO operator, the flux $\hat{f}_{j+\frac{1}{2}}$ requires a six point stencil $\{x_m\}_{m=j-3}^{j+3}$ so up to three ghost points are needed near the boundaries. We concentrate on describing how to define the values at the ghost points $\{x_j\}_{j=-3}^{-1}$ and $\{x_j\}_{j=N+1}^{N+3}$.

Our goal is to obtain spatial derivatives of each order at the physical boundary, then use Taylor expansion to get the values on the ghost points. The Taylor expansion of k th order at the left and right boundaries are respectively defined as

$$\begin{aligned} u_j &= \sum_{m=0}^k \frac{(x_j - a)^m}{m!} \partial_x^{(m)} u, & j = -3, -2, -1, \\ u_j &= \sum_{m=0}^k \frac{(x_j - b)^m}{m!} \partial_x^{(m)} u, & j = N+1, N+2, N+3. \end{aligned} \quad (2.4)$$

Here $\partial_x^{(m)} u$ is the numerical approximation of $\frac{\partial^m}{\partial x^m} u$ at the physical boundary. In fact, by repeatedly utilizing the equation (2.1) and the boundary condition at $x = a$, termed by *inverse Lax-Wendroff*, we can obtain the spatial derivatives as follows.

$$\begin{aligned} \partial_x^{(0)} u &= u(a, t) = g(t), \\ \partial_x^{(1)} u &= u_x(a, t) = -\frac{u_t}{f'(u)} \Big|_{x=a} = -\frac{g'(t)}{f'(g(t))}, \\ \partial_x^{(2)} u &= u_{xx}(a, t) = \frac{u_{tt}f'(u) - 2u_t^2 f''(u)}{f'(u)^3} \Big|_{x=a} = \frac{g''(t)f'(g(t)) - 2g'(t)^2 f''(g(t))}{f'(g(t))^3}, \\ &\vdots \end{aligned} \quad (2.5)$$

Thus we can obtain $\{\partial_x^{(m)} u\}_{m=0}^{\infty}$ completely from the given boundary condition and the PDE by converting the spatial derivatives into the time derivatives. Then, with Taylor expansion (2.4), we can obtain $\{u_j\}_{j=-3}^{-1}$. Notice that, if all the spatial derivatives are obtained by the ILW procedure, the values at the inflow ghost points are obtained completely from the

given boundary condition, no extrapolation from inside the computational domain is needed. Therefore, stability can be easily proved using the classical GKS theory [6].

Due to the heavy algebra of the ILW approach for the multi-dimensional nonlinear systems, a simplified ILW was proposed in [28], in which they obtained the first order derivative by the ILW procedure, and higher order derivatives by extrapolation. Theoretical stability analysis of upwind-biased finite difference schemes for linear conservation laws is reported in [13], and in it the authors showed the smallest number of derivatives which must be obtained by ILW rather than by extrapolation to ensure the stability under the maximum CFL condition of the internal scheme.

Glancing at the expressions of the derivatives in (2.5), we can immediately find that $f'(u)$ is in the denominators in the first and higher order derivatives. This is why the authors make the requirement that $f'(u)$ is away from zero on the boundary in [25]. If $f'(u)$ is exactly zero, it can be treated as outflow and the ILW procedure would work without any problem. This is the situation for the solid wall boundary conditions for Euler equations, which we will treat both in 1D and in 2D in the numerical results section. The difficulty is when the boundary condition, which is time dependent, transits from inflow to outflow (i.e. going through a sonic point). When $f'(u)$ is still positive but very small (for the left boundary, this is still an inflow), if we use the original ILW procedure, we would encounter a very small number in the denominator, which makes the algorithm much less robust, it could even blow up if we are unlucky and the denominator is too small. The main objective of this paper is to find an alternative ILW procedure which avoids small denominators near sonic points. Consider the one-dimensional scalar conservation laws (2.1) and the corresponding conservative scheme (2.3). The construction of the numerical flux $\hat{f}_{j+\frac{1}{2}}$ often involves a wide stencil near x_j . In fact, the Lax-Friedrichs flux in the fifth order WENO operator relies on $\{u_m\}_{m=j-3}^{j+3}$ and $\{f(u_m)\}_{m=j-3}^{j+3}$. Therefore, to make the scheme (2.3) work, we need not only the values $\{u_j\}_{j=0}^N$, but also the ghost point values $\{u_j\}_{j=-3}^{-1}$ and $\{u_j\}_{j=N+1}^{N+3}$. The traditional ILW procedure in [25] successfully obtained these values, as described above, then $\{f(u_j)\}_{j=-3}^{-1}$ and $\{f(u_j)\}_{j=N+1}^{N+3}$ are obtained immediately and the numerical flux can be formed. However, (2.5) shows that the ILW cannot tackle robustly with the case when $f'(u)$ is very close to zero on the boundary.

Inflow boundary: For the inflow boundary, we would insist on using the equation but avoiding $f'(u)$ appearing in the denominator. Our approach is to redefine $\{f_j\}_{j=-3}^{-1}$, where f_j is the approximation of $f(u)$ at x_j but is not taken simply as $f(u_j)$. A simple truncation error analysis shows that $O(\Delta x^5)$ difference between this treatment and the original ILW method in [25].

$$\begin{aligned}
f(u_j) &= f\left(\sum_{l=0}^4 \frac{(s_d)^l}{l!} \partial_x^{(l)} u\right) \\
&= f(u) + s_d f(u)_x + \frac{(s_d)^2}{2} f(u)_{xx} + \frac{(s_d)^3}{6} f(u)_{xxx} + \frac{(s_d)^4}{24} f(u)_{xxxx} + O(\Delta x^5),
\end{aligned} \tag{2.6}$$

where $j = -3, -2, -1$, $s_d = x_j - a$. Besides, the simplified ILW proposed in [28] suggests that, for the fifth order finite difference WENO scheme, we just need to obtain the first derivative by the ILW procedure and the second and higher order derivatives are obtained by extrapolation. Thus, we obtain $f(u)$ by the boundary condition, and $f(u)_x$ by the ILW approach, and $\{\partial_x^{(m)} f(u)\}_{m=2}^4$ by the extrapolation of the interior points. To further illustrate our idea, we show the fifth order treatment in the following.

$$\begin{aligned}
\partial_x^{(0)} f(u) &= f(g(t)), \\
\partial_x^{(1)} f(u) &= -g'(t), \\
\partial_x^{(2)} f(u) &= \frac{1}{\Delta x^2} \left(\frac{\delta_1^2}{2} (f(u_0) - 4f(u_1) + 6f(u_2) - 4f(u_3) + f(u_4)) \right. \\
&\quad + \frac{\delta_1}{2} (5f(u_0) - 18f(u_1) + 24f(u_2) - 14f(u_3) + 3f(u_4)) \\
&\quad \left. + \frac{1}{12} (35f(u_0) - 104f(u_1) + 114f(u_2) - 56f(u_3) + 11f(u_4)) \right), \\
\partial_x^{(3)} f(u) &= \frac{1}{\Delta x^3} \left(-\delta_1 (f(u_0) - 4f(u_1) + 6f(u_2) - 4f(u_3) + f(u_4)) \right. \\
&\quad \left. + \frac{1}{2} (-5f(u_0) + 18f(u_1) - 24f(u_2) + 14f(u_3) - 3f(u_4)) \right), \\
\partial_x^{(4)} f(u) &= \frac{1}{\Delta x^4} (f(u_0) - 4f(u_1) + 6f(u_2) - 4f(u_3) + f(u_4)).
\end{aligned} \tag{2.7}$$

Notice that in (2.7) we avoid placing $f'(u)$ in the denominator in obtaining $\{\partial_x^{(m)} f(u)\}_{m=0}^4$, thus we can define $\{f_j\}_{j=-3}^{-1}$ by using the Taylor expansion with $\{\partial_x^{(m)} f(u)\}_{m=0}^4$ even when $f'(u) = 0$. To obtain the values $\{u_j\}_{j=-3}^{-1}$, we only use the boundary condition and the extrapolation of interior points. We present the fifth order treatment as an example to show

how to obtain $\{\partial_x^{(m)}u\}_{m=0}^4$ at the left boundary.

$$\begin{aligned}
\partial_x^{(0)}u &= g(t), \\
\partial_x^{(1)}u &= \frac{1}{\Delta x} \left(-\frac{\delta_1^3}{6}(u_0 - 4u_1 + 6u_2 - 4u_3 + u_4) - \frac{\delta_1^2}{4}(5u_0 - 18u_1 + 24u_2 - 14u_3 + 3u_4) \right. \\
&\quad \left. + \frac{\delta_1}{12}(35u_0 - 104u_1 + 114u_2 - 56u_3 + 11u_4) \right. \\
&\quad \left. + \frac{1}{12}(-25u_0 + 48u_1 - 36u_2 + 16u_3 - 3u_4) \right), \\
\partial_x^{(2)}u &= \frac{1}{\Delta x^2} \left(\frac{\delta_1^2}{2}(u_0 - 4u_1 + 6u_2 - 4u_3 + u_4) + \frac{\delta_1}{2}(5u_0 - 18u_1 + 24u_2 - 14u_3 + 3u_4) \right. \\
&\quad \left. + \frac{1}{12}(35u_0 - 104u_1 + 114u_2 - 56u_3 + 11u_4) \right), \\
\partial_x^{(3)}u &= \frac{1}{\Delta x^3} \left(-\delta_1(u_0 - 4u_1 + 6u_2 - 4u_3 + u_4) + \frac{1}{2}(-5u_0 + 18u_1 - 24u_2 + 14u_3 - 3u_4) \right), \\
\partial_x^{(4)}u &= \frac{1}{\Delta x^4}(u_0 - 4u_1 + 6u_2 - 4u_3 + u_4).
\end{aligned} \tag{2.8}$$

Outflow boundary: For the outflow boundary, we obtain $\{u_j\}_{j=N+1}^{N+3}$ by extrapolation, and $f_j = f(u_j)$, $j = N+1, N+2, N+3$. If the solution of (2.1) is smooth, we can use Lagrange extrapolation to obtain these values. In this situation, the treatment is simply using the interior point values to construct a polynomial, and then extrapolate to the boundary. But if there is a shock near the boundary, then it may not have enough points between the shock and the boundary for high order extrapolation. To overcome this difficulty, in [25] Tan and Shu developed the WENO type extrapolation which would degenerate automatically to the lower order extrapolation but is more robust when the shock is near the boundary, while it maintains high order accuracy if the solution stays smooth near the boundary. Again, we first briefly review the procedure of the WENO type extrapolation proposed in [25], and take the fifth order treatment as the illustration example. Now our goal is to obtain a $(5-m)$ th order approximation of $\frac{\partial^m}{\partial x^m}u$ on the boundary. Assume we have five candidate stencils given by

$$S_r = \{x_{N-r}, \dots, x_N\}, \quad r = 0, \dots, 4.$$

Then we can construct the Lagrange polynomials of degree r on $\{S_r\}_{r=0}^4$, denoted as $\{p_m(x)\}_{m=0}^4$. Suppose u is smooth on S_4 , by the approximation theory we have $u = p_m + O(\Delta x^{m+1})$, $m = 0, \dots, 4$. This indicates that $\sum_{r=0}^4 d_r p_r(x)$ is a fifth order approximation to u where

$d_0 = \Delta x^4$, $d_1 = \Delta x^3$, $d_2 = \Delta x^2$, $d_3 = \Delta x$, $d_4 = 1 - \sum_{r=0}^3 d_r$. Then we take

$$\partial_x^{(m)} u = \frac{d^m}{dx^m} \left(\sum_{r=0}^4 d_r p_r(x) \right) = \sum_{r=0}^4 d_r \frac{d^m}{dx^m} p_r(x) \Big|_{x=b},$$

such that $\partial_x^{(m)} u$ is $(5 - m)$ -th order approximation to $\frac{\partial^m}{\partial x^m} u$. To obtain a WENO type extrapolation, the coefficients $\{d_r\}_{r=0}^4$ are changed into $\{\omega_r\}_{r=0}^4$, where

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^4 \alpha_s}, \quad \alpha_r = \frac{d_r}{(\varepsilon + \beta_r)^q}, \quad (2.9)$$

with $\varepsilon = 10^{-6}$ and β_r are the smoothness indicators given in the following.

$$\beta_0 = \Delta x^2, \quad \beta_m = \sum_{l=1}^4 \Delta x^{2l-1} \int_{x_{N-1}}^{x_{N+1}} \left(\frac{d^l}{dx^l} p_m(x) \right)^2 dx, \quad m = 1, \dots, 4. \quad (2.10)$$

The WENO type extrapolation works very well in [25, 26, 28, 12], etc. However, it is preferable to preserve the property of self-similarity, which is intrinsic to the hyperbolic conservation law but is not fulfilled by this kind of WENO type extrapolation because the nonlinear weights ω_r depend on the mesh size Δx explicitly. Besides, there is a parameter q in the smoothness indicator (2.9) and its value is problem dependent in [25, 28]. We would prefer to obtain an extrapolation which preserves the property of self-similarity when computing the hyperbolic conservation laws.

In the following we present the new WENO type extrapolation. We will adopt the above notations without any ambiguities. We still perform the construction on the five-point stencil $S = \{x_{N-4}, \dots, x_N\}$ with the idea of the multi-resolution WENO methods in [32]. Assume we have the point values $\{v_{N-4}, \dots, v_N\}$ of some function $v(x)$. We then have the five sub-stencils

$$S_r = \{x_{N-r}, \dots, x_N\}, \quad r = 0, \dots, 4,$$

and the corresponding interpolation polynomials are denoted as $q_r(x)$, $r = 0, \dots, 4$. If $v(x)$ is smooth on the stencil S_4 , then $q_4(x)$ is the desired polynomial. If $v(x)$ has a discontinuity in the interval (x_{N-4}, x_{N-3}) but smooth in (x_{N-3}, x_N) , then the polynomial $q_3(x)$ is desired. The cases that the discontinuity located in (x_{N-3}, x_{N-2}) , (x_{N-2}, x_{N-1}) and (x_{N-1}, x_N) are similar. In summary, we must use the polynomial that includes the point value v_N , and the stencil should be chosen as large as possible and the function $v(x)$ stays smooth on it in the meantime. In the following we give a detailed description on how to achieve this goal, while the key idea comes from [32].

We first present the equivalent expressions of $q_r(x)$ as follows.

$$p_0(x) = q_0(x), \quad p_r(x) = \sum_{m=0}^r \frac{d_m}{d_r} q_r(x) - \sum_{m=0}^{r-1} \frac{d_m}{d_r} p_m(x), \quad r = 1, \dots, 4, \quad (2.11)$$

where $\{d_r\}_{r=0}^4$ satisfy $\sum_{r=0}^4 d_r = 1$, $d_r > 0$, $r = 0, \dots, 4$. Then $\{d_r\}_{r=0}^4$ are the linear weights since we have $q_4(x) = \sum_{r=0}^4 d_r p_r(x)$. Throughout this paper we take $\{d_r\}_{r=0}^4$ as follows.

$$d_0 = \frac{1}{15}, \quad d_1 = \frac{2}{15}, \quad d_2 = \frac{1}{5}, \quad d_3 = \frac{4}{15}, \quad d_4 = \frac{1}{3}. \quad (2.12)$$

Similar to the WENO-Z idea in [1], the nonlinear weights are taken in the following form.

$$\begin{aligned} \omega_r &= \frac{\alpha_r}{\sum_{s=0}^4 \alpha_s}, \quad \alpha_r = d_r \left(1 + \left(\frac{\tau}{\varepsilon + \beta_r} \right)^4 \right), \quad r = 0, \dots, 4, \\ \tau &= \left(\max_{1 \leq l \leq 3} \{(\beta_l - \beta_4)^2\} \right)^{\frac{1}{2}} + \max_{1 \leq l \leq 4} \{\|q_0(x) - q_l(x)\|^3\} \end{aligned} \quad (2.13)$$

where the $\|\cdot\|$ is the standard L^2 -norm on (x_{N-1}, x_{N+1}) , and the smoothness indicators $\{\beta_r\}_{r=0}^4$ are defined as follows.

$$\begin{aligned} \beta_0 &= c_0 \beta_1, \\ \beta_r &= \sum_{l=1}^r \Delta x^{2l-1} \int_{x_{N-1}}^{x_{N+1}} \left(\frac{d^l}{dx^l} q_r(x) \right)^2 dx, \quad r = 1, \dots, 4, \end{aligned} \quad (2.14)$$

with $\varepsilon = 10^{-4}$ is placed to avoid the denominator becoming zero (notice that this choice of ε would make the denominator to become $(10^{-4})^4 = 10^{-16}$ when β_r is zero or is very small, which is close to machine zero in double precision), and c_0 is a positive constant. Note that $p_0(x)$ is a constant function so its smoothness indicator vanishes by the second equation in (2.14), thus we take β_0 be proportional to β_1 . When c_0 is small enough, the extrapolation tends to be the constant extrapolation when there is a discontinuity locates on (x_{N-2}, x_N) . Throughout this paper, we take $c_0 = 0.1$ and later we can see it works well in the numerical tests. With this new WENO type extrapolation, we have the extrapolating polynomial as

$$p(x) = \sum_{r=0}^4 \omega_r p_r(x).$$

With the newly obtained polynomial $p(x)$, we have the derivatives $\{\partial_x^{(r)} u\}_{r=0}^4$ at $x = b$.

$$\partial_x^{(r)} u = \frac{d^r}{dx^r} p(x), \quad r = 0, \dots, 4. \quad (2.15)$$

With these derivatives on $x = b$, we then use Taylor expansion (2.4) to obtain the values on the ghost points $\{x_j\}_{j=N+1}^{N+3}$.

Remark 2.1. *We have made two modifications from the previous algorithms in [25, 28]. One is to separate the evaluation of u and $f(u)$ on the ghost points, thus it allows us to handle the case when $f'(u)$ vanishes on the boundary. The other is to make use of the multi-resolution WENO procedure to obtain a new WENO type extrapolation, which preserves the property of self-similarity. This is because we use the constants in the linear weights in (2.13), thus it is independent of Δx . Also, the second and higher order derivatives are obtained by extrapolation, and we still need to use the new WENO type extrapolation when the shock comes near the boundary.*

2.2 One-dimensional systems

Consider the 1D compressible Euler equations

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x = 0, \quad x \in (a, b), \quad t > 0, \quad (2.16)$$

where $\mathbf{U}$ and $\mathbf{F}(\mathbf{U})$ are defined as

$$\mathbf{U} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix}.$$

ρ, u, p and E stand for density, velocity, pressure and total energy per unit volume, respectively. The equation of state is

$$E = p/(\gamma - 1) + \rho u^2/2,$$

where γ is the heat capacity ratio and $\gamma = 1.4$ for air when the temperature is within a suitable range. Notice that U_1 no longer stands for the value of U at $x = x_1$, but only stands for the density ρ . So are U_2 and U_3 . Thus, we take the notation $(U_1)_j$ to stand for the value of U_1 at $x = x_j$, and $\mathbf{U}_j = ((U_1)_j, (U_2)_j, (U_3)_j)^T$. Similarly, $(F_1)_j$ is the value of $F_1(\mathbf{U})$ at $x = x_j$, and $\mathbf{F}_j = (F_1(\mathbf{U}_j), F_2(\mathbf{U}_j), F_3(\mathbf{U}_j))^T$. Without loss of generality, we consider the left boundary $x = a$. Firstly, we rewrite the governing equation (2.16) into the following form.

$$\mathbf{U}_t + \mathbf{F}'(\mathbf{U})\mathbf{U}_x = 0,$$

where the Jacobian matrix $\mathbf{F}'(\mathbf{U})$ is given as

$$\mathbf{F}'(\mathbf{U}) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma - 3)u^2 & (3 - \gamma)u & \gamma - 1 \\ \frac{1}{2}(\gamma - 1)u^3 - uH & H - (\gamma - 1)u^2 & \gamma u \end{pmatrix}, \quad (2.17)$$

with the enthalpy $H = (E+p)/\rho$. The number of the boundary conditions are determined by the signs of the eigenvalues of the Jacobian matrix $\mathbf{F}'(\mathbf{U})$. By the similarity transformation $\mathbf{F}'(\mathbf{U}) = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$, we have

$$\mathbf{V}_t + \mathbf{\Lambda}\mathbf{V}_x = 0, \quad (2.18)$$

where $\mathbf{V} = \mathbf{R}^{-1}\mathbf{U}$ are characteristic variables, $\mathbf{\Lambda} = \text{diag}(u-c, u, u+c)$, and $c = \sqrt{\gamma p/\rho}$ is the speed of sound, and $\mathbf{R}$ and $\mathbf{R}^{-1}$ are given as

$$\begin{aligned} \mathbf{R} &= \begin{pmatrix} 1 & 1 & 1 \\ u-c & u & u+c \\ H-uc & \frac{1}{2}u^2 & H+uc \end{pmatrix}, \\ \mathbf{R}^{-1} &= \frac{1}{c^2} \begin{pmatrix} \frac{1}{2}uc + \frac{1}{4}(\gamma-1)u^2 & -\frac{1}{2}(\gamma-1)u - \frac{1}{2}c & \frac{1}{2}(\gamma-1) \\ c^2 - \frac{1}{2}(\gamma-1)u^2 & (\gamma-1)u & 1-\gamma \\ -\frac{1}{2}uc + \frac{1}{4}(\gamma-1)u^2 & -\frac{1}{2}(\gamma-1)u + \frac{1}{2}c & \frac{1}{2}(\gamma-1) \end{pmatrix}. \end{aligned} \quad (2.19)$$

In the finite difference WENO scheme, it takes this characteristic decomposition when obtaining the numerical fluxes, and now we use this decomposition at $x = a$ to determine the inflow boundary conditions and outflow boundary conditions.

Now let us consider four cases in the following:

Case 1: $u - c > 0$;

Case 2: $u - c \leq 0, \quad u > 0$;

Case 3: $u \leq 0, \quad u + c > 0$;

Case 4: $u + c \leq 0$.

Note that the above u and c in these four cases are obtained at $x = a$.

For the case 1, we have all eigenvalues positive, thus we need three inflow boundary conditions. Our goal is to obtain both $\{\partial_x^{(l)}\mathbf{U}\}_{l=0}^4$ and $\{\partial_x^{(l)}\mathbf{F}(\mathbf{U})\}_{l=0}^4$ at $x = a$, then we can use the Taylor expansion (2.4) to obtain $\mathbf{U}$ and $\mathbf{F}(\mathbf{U})$ on the ghost points $\{x_j\}_{j=-3}^{-1}$. For simplicity, we denote $(w)^{ilw}$ if w is obtained from the boundary conditions and the governing equations, and $(w)^{ext}$ if w is obtained from the extrapolation of the interior points. Similar to the scalar case in (2.8), we first consider $\mathbf{U} = (U_1, U_2, U_3)^T$ and use the boundary conditions $(\partial_x^{(0)}\mathbf{U})^{ilw}$ and $\{(\partial_x^{(l)}\mathbf{U})^{ext}\}_{l=1}^4$ to obtain $\{\partial_x^{(l)}\mathbf{U}\}_{l=0}^4$ at $x = a$.

$$\partial_x^{(0)}\mathbf{U} = (\mathbf{U})^{ilw}, \quad \partial_x^{(l)}\mathbf{U} = (\partial_x^{(l)}\mathbf{U})^{ext}, \quad l = 1, \dots, 4. \quad (2.20)$$

By the Taylor expansion (2.4), we obtain $\{\mathbf{U}_j\}_{j=-3}^{-1}$. Then we consider $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$. Similar to the scalar case in (2.7), we use the boundary conditions $\partial_x^{(0)} \mathbf{U}$ to obtain $\partial_x^{(0)} \mathbf{F}(\mathbf{U})$, obtain $\partial_x^{(1)} \mathbf{F}(\mathbf{U})$ by the ILW procedure, and obtain $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=2}^4$ by extrapolation at $x = a$.

$$\begin{aligned}\partial_x^{(0)} \mathbf{F}(\mathbf{U}) &= \mathbf{F}(\partial_x^{(0)} \mathbf{U}), \quad \partial_x^{(1)} \mathbf{F}(\mathbf{U}) = (\partial_x^{(1)} \mathbf{F}(\mathbf{U}))^{ilw}, \\ \partial_x^{(l)} \mathbf{F}(\mathbf{U}) &= (\partial_x^{(l)} \mathbf{F}(\mathbf{U}))^{ext}, \quad l = 2, 3, 4.\end{aligned}\tag{2.21}$$

By the Taylor expansion (2.4), we obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$. Then case 1 is finished.

For the case 2, we have two positive eigenvalues and one negative eigenvalue, thus we need two boundary conditions at $x = a$. For convenience, we assume that the two boundary conditions are $\rho(a, t) = g_1(t)$, $\rho u(a, t) = g_2(t)$. In fact, it is equivalent to prescribing the incoming characteristic variable V_2, V_3 as a function of the outgoing characteristic variable V_1 , where $(V_1, V_2, V_3)^T = \mathbf{V} = \mathbf{R}^{-1} \mathbf{U}$ [25].

As before, we shall obtain $\mathbf{U}$ on the boundary $x = a$ firstly. With the extrapolation and the prescribed boundary conditions, we can obtain $\mathbf{U}$ at $x = a$, i.e. $U_1 = g_1(t), U_2 = g_2(t), U_3 = (U_3)^{ext}$. With $\mathbf{U}$ at $x = a$, we are able to perform the local characteristic decomposition and obtain the incoming characteristic variables V_2, V_3 and outgoing variable V_1 in the interior domain. In particular, the outgoing characteristic variable V_1 is used for extrapolation. For simplicity we denote $\mathbf{R} = (r_{ij})_{3 \times 3}$, $\mathbf{R}^{-1} = (\tilde{r}_{ij})_{3 \times 3}$. Then at the left boundary $x = a$, we have

$$\begin{aligned}\partial_x^{(0)} U_1 &= g_1(t), \quad \partial_x^{(0)} U_2 = g_2(t), \quad \partial_x^{(0)} U_3 = ((V_1)^{ext} - \tilde{r}_{11} g_1(t) - \tilde{r}_{21} g_2(t)) / \tilde{r}_{23}, \\ \partial_x^{(l)} U_1 &= (\partial_x^{(l)} U_1)^{ext}, \quad \partial_x^{(l)} U_2 = (\partial_x^{(l)} U_2)^{ext}, \quad l = 1, \dots, 4.\end{aligned}\tag{2.22}$$

From (2.19) we know that $\tilde{r}_{13} = (\gamma - 1)/(2c^2) \neq 0$, thus (2.22) is well-defined. In (2.22) $\partial_x^{(0)} U_3$ obtained on the boundary is not for the Taylor expansion, but for computing $\mathbf{F}(\mathbf{U})$ on the boundary. Now we assume $\{(U_1)_j\}_{j=-3}^{-1}$ and $\{(U_2)_j\}_{j=-3}^{-1}$ are obtained by (2.22) and (2.4). To obtain $\{(U_3)_j\}_{j=-3}^{-1}$, with the relation $\mathbf{V} = \mathbf{R}^{-1} \mathbf{U}$ we have

$$U_3 = ((V_1)^{ext} - \tilde{r}_{12} U_1 - \tilde{r}_{13} U_2) / \tilde{r}_{13}.\tag{2.23}$$

Notice that (2.22) is performed at the left boundary $x = a$ while (2.23) is considered on the ghost points $\{x_j\}_{j=-3}^{-1}$.

Now let us turn to the definitions of $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$. Since we have obtained $\partial_x^{(0)} \mathbf{U}$ at the left boundary $x = a$, we immediately obtain $\partial_x^{(0)} \mathbf{F}(\mathbf{U})$ with the expression $\mathbf{F}(\partial_x^{(0)} \mathbf{U})$. The key step in this algorithm is to obtain $\partial_x^{(1)} \mathbf{F}(\mathbf{U})$ by the ILW procedure. By the boundary conditions and the governing equations, we can obtain $\partial_x^{(1)} F_1(\mathbf{U})$ and $\partial_x^{(1)} F_2(\mathbf{U})$. To obtain

$\partial_x^{(1)} F_3(\mathbf{U})$, we use the relation $\mathbf{R}^{-1} \mathbf{F}(\mathbf{U})_x = \mathbf{\Lambda} \mathbf{V}_x$. The second order and higher order derivatives are obtained by extrapolation. We summarize the procedure in the following.

$$\begin{aligned}\partial_x^{(0)} \mathbf{F}(\mathbf{U}) &= \mathbf{F}(\partial_x^{(0)} \mathbf{U}), \\ \partial_x^{(1)} F_1(\mathbf{U}) &= (\partial_x^{(1)} F_1)^{ilw} = -g'_1(t), \quad \partial_x^{(1)} F_2(\mathbf{U}) = (\partial_x^{(1)} F_2)^{ilw} = -g'_2(t), \\ \partial_x^{(1)} F_3(\mathbf{U}) &= ((u-c)(\partial_x^{(1)} V_1)^{ext} + \tilde{r}_{11} g'_1(t) + \tilde{r}_{12} g'_2(t)) / \tilde{r}_{13}, \\ \partial_x^{(l)} \mathbf{F}(\mathbf{U}) &= (\partial_x^{(l)} \mathbf{F}(\mathbf{U}))^{ext}, \quad l = 2, 3, 4.\end{aligned}\tag{2.24}$$

By the Taylor expansion (2.4), we obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$. Then case 2 is finished.

For the case 3, we have one inflow boundary condition and two outflow boundary conditions. Assume we have the boundary condition for $\rho(x, t) = g_1(t)$. Firstly we use the extrapolation and boundary condition to obtain $\mathbf{U}$ at $x = a$. Then we perform the characteristic decomposition at $x = a$ and obtain the characteristic variables $\mathbf{V}$, and V_1 and V_2 are outgoing characteristic variables and V_3 is incoming characteristic variable. By the relation $\mathbf{V} = \mathbf{R}^{-1} \mathbf{U}$, we have

$$\begin{aligned}\partial_x^{(0)} \mathbf{U} &= \mathbf{R} \mathbf{V}, \text{ where } V_1 = (V_1)^{ext}, V_2 = (V_2)^{ext}, V_3 = (g_1(t) - r_{11}(V_1)^{ext} - r_{12}(V_2)^{ext}) / r_{13}, \\ \partial_x^{(l)} U_1 &= (\partial_x^{(l)} U_1)^{ext}, \quad l = 1, \dots, 4,\end{aligned}\tag{2.25}$$

where $r_{13} = 1 \neq 0$ thus V_3 is well-defined in (2.25). By the Taylor expansion (2.4), we obtain $\{(U_1)_j\}_{j=-3}^{-1}$. To obtain $\{(U_2)_j\}_{j=-3}^{-1}$ and $\{(U_3)_j\}_{j=-3}^{-1}$, we have $\{(\partial_x^{(l)} V_1)^{ext}\}_{l=0}^4$ and $\{(\partial_x^{(l)} V_2)^{ext}\}_{l=0}^4$ at the boundary and we use Taylor expansion (2.4) to obtain $\{(V_1)_j\}_{j=-3}^{-1}$ and $\{(V_2)_j\}_{j=-3}^{-1}$. Then we use the relation $\mathbf{V} = \mathbf{R}^{-1} \mathbf{U}$ to obtain U_2 and U_3 .

$$\mathbf{U} = \mathbf{R} \mathbf{V}, \text{ where } V_1 = (V_1)^{ext}, V_2 = (V_2)^{ext}, V_3 = (U_1 - r_{11}(V_1)^{ext} - r_{12}(V_2)^{ext}) / r_{13}.\tag{2.26}$$

Notice that (2.25) is performed at the left boundary $x = a$ while (2.26) is considered on the ghost points $\{x_j\}_{j=-3}^{-1}$.

In the following we consider $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$. In the case 2 we have elaborated on how to obtain $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ and case 3 follows almost in the same way. Thus we present the algorithm in the following directly.

$$\begin{aligned}\partial_x^{(0)} \mathbf{F}(\mathbf{U}) &= \mathbf{F}(\partial_x^{(0)} \mathbf{U}), \\ \partial_x^{(1)} \mathbf{F}(\mathbf{U}) &= \mathbf{R} \mathbf{\Lambda} \mathbf{V}_x, \text{ where } (u-c)\partial_x^{(1)} V_1 = (u-c)(\partial_x^{(1)} V_1)^{ext}, \quad u\partial_x^{(1)} V_2 = u(\partial_x^{(1)} V_2)^{ext}, \\ (u+c)\partial_x^{(1)} V_3 &= (-g'_1(t) - r_{11}(u-c)\partial_x^{(1)} V_1 - r_{12}u\partial_x^{(1)} V_2) / r_{13}, \\ \partial_x^{(l)} \mathbf{F}(\mathbf{U}) &= (\partial_x^{(l)} \mathbf{F}(\mathbf{U}))^{ext}, \quad l = 2, 3, 4.\end{aligned}\tag{2.27}$$

The ILW procedure in (2.27) is in the third line, in which we use the relation $\partial_x^{(1)} F_1(\mathbf{U}) = (\partial_x^{(1)} F_1)^{ilw} = -g'_1(t)$ in obtaining $(u+c)\partial_x^{(1)} V_3$. By the Taylor expansion (2.4), we obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$. Then case 3 is finished.

For the case 4, we have three outflow boundary conditions. The treatment of this case is very simple. Firstly, we have the characteristic decomposition at $x = a$ and we can obtain the characteristic variables $\{\partial_x^{(l)} \mathbf{V}\}_{l=0}^4$ at $x = a$. With Taylor expansion (2.4), we then obtain $\mathbf{V}$ and transform $\mathbf{V}$ into $\mathbf{U}$ by the relation $\mathbf{U} = \mathbf{R}\mathbf{V}$, therefore we obtain $\{\mathbf{U}_j\}_{j=-3}^{-1}$.

$$\mathbf{U} = \mathbf{R}(\mathbf{V})^{ext}. \quad (2.28)$$

After we obtain the $\{\mathbf{U}_j\}_{j=-3}^{-1}$, we substitute them into $\mathbf{F}(\mathbf{U})$ and obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$, thus case 4 is finished.

We summarize our algorithm of the boundary treatment at the left boundary as follows. Assume we have obtained $\{\mathbf{U}_j\}_{j=0}^N$ at time level t_n . Our goal is to obtain $\{\mathbf{U}_j\}_{j=-3}^{-1}$, $\{\mathbf{F}_j\}_{j=-3}^{-1}$.

1. Firstly, obtain $\mathbf{U}$ with the extrapolation and boundary conditions at the boundary. Perform the characteristic decomposition, and decide the prescribed inflow boundary conditions according to the signs of the eigenvalues of $\mathbf{\Lambda}$ in (2.18).
2. There are four cases of the different signs of eigenvalues of $\mathbf{\Lambda}$ and they need different treatments.
 - Case 1: $u - c > 0$. We have three inflow boundary conditions at the boundary $x = a$. We obtain $\{\partial_x^{(l)} \mathbf{U}\}_{l=0}^4$ by (2.20), then we can obtain $\{\mathbf{U}_j\}_{j=-3}^{-1}$ by the Taylor expansion (2.4). Also, we have $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ by (2.21), and we can obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$ by the Taylor expansion (2.4).
 - Case 2: $u - c \leq 0$, $u > 0$. We have two inflow boundary conditions and one outflow boundary condition at the boundary $x = a$. We firstly obtain $\{\partial_x^{(l)} U_1\}_{l=0}^4$ and $\{\partial_x^{(l)} U_2\}_{l=0}^4$ by (2.22), then we can obtain U_1 and U_2 at $x = x_{-3}, x_{-2}, x_{-1}$ by the Taylor expansion (2.4). With the extrapolation of V_1 , we can obtain U_3 at $x = x_{-3}, x_{-2}, x_{-1}$ by (2.23). Then we have $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ by (2.24), and we can obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$ by the Taylor expansion (2.4).
 - Case 3: $u \leq 0$, $u + c > 0$. We have one inflow boundary condition and two outflow boundary conditions at the boundary $x = a$. We firstly obtain $\{\partial_x^{(l)} U_1\}_{l=0}^4$ by (2.25), then we can obtain U_1 at $x = x_{-3}, x_{-2}, x_{-1}$ by the Taylor expansion (2.4). With extrapolation of V_1 and V_2 , we can obtain U_2 and U_3 at $x = x_{-1}, x_{-2}, x_{-3}$

by (2.26). We then have $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ by (2.27), and we can obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$ by the Taylor expansion (2.4).

- Case 4: $u + c \leq 0$. We have three outflow boundary conditions at the boundary $x = a$. by the extrapolation of $\mathbf{V}$, we can obtain $\mathbf{U}$ at $\{x_j\}_{j=-3}^{-1}$ by (2.28). And we can obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$ by substitute $\mathbf{U}$ into $\mathbf{F}(\mathbf{U})$ at $\{x_j\}_{j=-3}^{-1}$.

Remark 2.2. *Many physical problems are described by the compressible inviscid Euler equations with the no-penetration boundary condition at solid walls. In the computation, the most popular way to impose the no-penetration boundary condition is the reflection technique, that all interior solution components are reflected symmetrically to the values of the ghost points, except for the normal velocity whose sign is reversed. For the treatment of the solid wall boundary condition $u = 0$, we would like to use the boundary condition $pu = 0$ on the boundary and adopt the case 3 in the above algorithm. In this case, the algorithm needs some changes because we prescribe the boundary condition for ρ as an illustration example in the above algorithm. We now show the modifications of the case 3 for treating the solid wall boundary condition and briefly write them down as follows.*

Assume we have the boundary condition $pu = 0$ at $x = a$. We obtain $\{\partial_x^{(l)} U_1\}_{l=0}^4$ by the following equations.

$$\begin{aligned} \partial_x^{(0)} \mathbf{U} &= \mathbf{R} \mathbf{V}, \text{ where } V_1 = (V_1)^{ext}, V_2 = (V_2)^{ext}, V_3 = (-r_{21}(V_1)^{ext} - r_{22}(V_2)^{ext})/r_{23}, \\ \partial_x^{(l)} U_2 &= (\partial_x^{(l)} U_2)^{ext}, l = 1, \dots, 4. \end{aligned} \quad (2.29)$$

By the Taylor expansion, we obtain $\{(U_2)_j\}_{j=-3}^{-1}$. With the extrapolation of V_1 and V_2 , we can obtain U_1 and U_3 on the ghost points $\{x_j\}_{j=-3}^{-1}$ by the following equations.

$$\mathbf{U} = \mathbf{R} \mathbf{V}, \text{ where } V_1 = (V_1)^{ext}, V_2 = (V_2)^{ext}, V_3 = (U_2 - r_{21}(V_1)^{ext} - r_{22}(V_2)^{ext})/r_{23}. \quad (2.30)$$

After we obtain $\partial_x^{(0)} \mathbf{U}$ on the ghost points $\{x_j\}_{j=-3}^{-1}$, we then have $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ at $x = a$ by the following equations.

$$\begin{aligned} \partial_x^{(0)} \mathbf{F}(\mathbf{U}) &= \mathbf{F}(\partial_x^{(0)} \mathbf{U}), \\ \partial_x^{(1)} \mathbf{F}(\mathbf{U}) &= \mathbf{R} \Lambda \mathbf{V}_x, \text{ where } (u - c) \partial_x^{(1)} V_1 = (u - c) (\partial_x^{(1)} V_1)^{ext}, u \partial_x^{(1)} V_2 = u (\partial_x^{(1)} V_2)^{ext}, \\ (u + c) \partial_x^{(1)} V_3 &= (-r_{21}(u - c) \partial_x^{(1)} V_1 - r_{22} u \partial_x^{(1)} V_2)/r_{23}, \\ \partial_x^{(l)} \mathbf{F}(\mathbf{U}) &= (\partial_x^{(l)} \mathbf{F}(\mathbf{U}))^{ext}, l = 2, 3, 4. \end{aligned} \quad (2.31)$$

With the derivatives $\{\partial_x^{(l)} \mathbf{F}(\mathbf{U})\}_{l=0}^4$ at $x = a$, we can obtain $\{\mathbf{F}_j\}_{j=-3}^{-1}$ by the Taylor expansion.

Remark 2.3. *If the solution is smooth near the boundary, then we can use Lagrange extrapolation in the above algorithm. But if there is a shock near the boundary, then in the above algorithm we recommend to use the WENO type extrapolation given in the 1D scalar case, which makes the algorithm more robust.*

2.3 Two-dimensional problems

In this subsection, we generalize the approach to the two-dimensional problems. The two-dimensional compressible Euler equations are given as

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = 0, \quad (x, y) \in \Omega, \quad t > 0, \quad (2.32)$$

where

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix},$$

with suitable boundary conditions and initial conditions. ρ, u, v, p and E stand for the density, x -velocity, y -velocity, pressure and total energy per unit volume, respectively. The equation of state is

$$E = p/(\gamma - 1) + \rho(u^2 + v^2)/2,$$

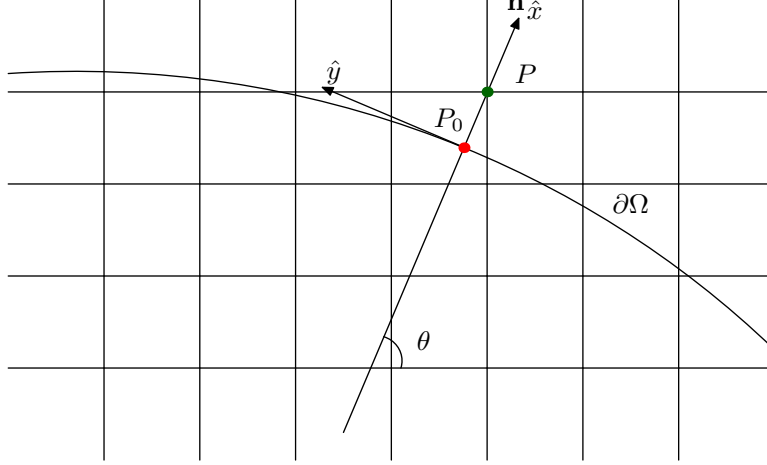
where γ is the heat capacity ratio and $\gamma = 1.4$ for air when the temperature is within a suitable range.

Assume we have a Cartesian mesh for Ω and the boundary $\partial\Omega$ may not be aligned with the grid points. Then the grid points inside the domain Ω are called the interior grid points. Assume we have the fifth order finite difference WENO method as our interior scheme and the values on the interior grid points are obtained already. Our goal is to obtain the values on the ghost points, denoted as P in Figure 1. Following [25], we first find the P_0 on $\partial\Omega$ so that the outward normal $\mathbf{n}$ to $\partial\Omega$ at P_0 goes through P .

We set up a local coordinate system at P_0 by

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \mathbf{T} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix}, \quad (2.33)$$

Figure 1: The local coordinate at P_0 in 2D problems: the ghost point P marked in green and its projection point $P_0 \in \partial\Omega$ marked in red.



where θ is the angle between the outward normal $\mathbf{n}$ to $\partial\Omega$ at P_0 and the x -axis, and $\mathbf{T}$ is the rotation matrix. Through this rotation, $\hat{x}$ -axis is aligned with the outward normal $\mathbf{n}$. Now we denote

$$\hat{\mathbf{U}} = (\hat{U}_1, \hat{U}_2, \hat{U}_3, \hat{U}_4)^T = (\rho, \rho\hat{u}, \rho\hat{v}, E)^T,$$

where $\hat{u}$ and $\hat{v}$ are given as

$$\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \mathbf{T} \begin{pmatrix} u \\ v \end{pmatrix},$$

then the Euler equations (2.32) can be transformed into the following form

$$\hat{\mathbf{U}}_t + \mathbf{F}(\hat{\mathbf{U}})_{\hat{x}} + \mathbf{G}(\hat{\mathbf{U}})_{\hat{y}} = 0. \quad (2.34)$$

We still use the following notations without causing any ambiguities.

$$\mathbf{F}(\hat{\mathbf{U}}) = (F_1, F_2, F_3, F_4)^T, \quad \mathbf{G}(\hat{\mathbf{U}}) = (G_1, G_2, G_3, G_4)^T.$$

Thus, at P_0 we will consider the transformed Euler equations (2.34) instead of the original one for convenience. Once we obtain $\hat{\mathbf{U}}$, we can transform back to $\mathbf{U}$ without any difficulties. We now show the transformation of $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$ from $\mathbf{F}(\hat{\mathbf{U}})$ and $\mathbf{G}(\hat{\mathbf{U}})$ as follows.

$$\begin{aligned} F_1(\mathbf{U}) &= T_{11}F_1 + T_{21}G_1, \\ F_2(\mathbf{U}) &= (T_{11})^2F_2 + 2T_{11}T_{21}F_3 + (T_{21})^2G_3, \\ F_3(\mathbf{U}) &= T_{11}T_{12}(F_2 - G_3) + (T_{11}T_{22} + T_{12}T_{21})F_3, \\ F_4(\mathbf{U}) &= T_{11}F_4 + T_{21}G_4. \end{aligned} \quad (2.35)$$

$$\begin{aligned}
G_1(\mathbf{U}) &= T_{12}F_1 + T_{22}G_1, \\
G_2(\mathbf{U}) &= T_{11}T_{12}(F_2 - G_3) + (T_{11}T_{22} + T_{12}T_{22})F_3, \\
G_3(\mathbf{U}) &= (T_{12})^2F_2 + 2T_{12}T_{22}F_3 + (T_{22})^2G_3, \\
G_4(\mathbf{U}) &= T_{12}F_4 + T_{22}G_4.
\end{aligned} \tag{2.36}$$

In the following we show how to obtain the values of $\mathbf{F}(\hat{\mathbf{U}})$ and $\mathbf{G}(\hat{\mathbf{U}})$ on the ghost point P . First we need the normal derivatives, i.e. the $\hat{x}$ -directional derivatives, up to 4-th order. Now we rewrite the transformed Euler equations (2.34) into the following form.

$$\hat{\mathbf{U}}_t + \mathbf{F}'(\hat{\mathbf{U}})\mathbf{U}_{\hat{x}} + \mathbf{G}(\hat{\mathbf{U}})_{\hat{y}} = 0,$$

where the Jacobian matrix $\mathbf{F}'(\hat{\mathbf{U}})$ is given as

$$\mathbf{F}'(\hat{\mathbf{U}}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{1}{2}(\gamma - 1)(\hat{u}^2 + \hat{v}^2) - \hat{u}^2 & (3 - \gamma)\hat{u} & (1 - \gamma)\hat{v} & \gamma - 1 \\ -\hat{u}\hat{v} & \hat{v} & \hat{u} & 0 \\ \frac{1}{2}(\gamma - 1)\hat{u}(\hat{u}^2 + \hat{v}^2) - \hat{u}H & H + (1 - \gamma)\hat{u}^2 & (1 - \gamma)\hat{u}\hat{v} & \gamma\hat{u} \end{pmatrix} \tag{2.37}$$

and $H = (E + p)/\rho$ is the enthalpy as in the one-dimensional case. With the similarity transformation $\mathbf{F}'(\hat{\mathbf{U}}) = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^{-1}$, we perform the characteristic decomposition and obtain

$$\mathbf{V}_t + \mathbf{\Lambda}\mathbf{V}_{\hat{x}} + \mathbf{R}^{-1}\mathbf{G}(\hat{\mathbf{U}})_{\hat{y}} = 0, \tag{2.38}$$

where the characteristic variables $\mathbf{V} = \mathbf{R}^{-1}\hat{\mathbf{U}}$, $\mathbf{\Lambda} = \text{diag}(\hat{u} - c, \hat{u}, \hat{u}, \hat{u} + c)$, $c = \sqrt{\gamma p/\rho}$ and

$$\begin{aligned}
\mathbf{R} &= \begin{pmatrix} 1 & 0 & 1 & 1 \\ \hat{u} - c & 0 & \hat{u} & \hat{u} + c \\ \hat{v} & 1 & \hat{v} & \hat{v} \\ H - \hat{u}c & \hat{v} & \frac{1}{2}(\hat{u}^2 + \hat{v}^2) & H + \hat{u}c \end{pmatrix}, \\
\mathbf{R}^{-1} &= \frac{1}{c^2} \begin{pmatrix} \frac{1}{2}(b_1 + \hat{u}c) & -\frac{1}{2}((\gamma - 1)\hat{u} + c) & -\frac{1}{2}(\gamma - 1)\hat{v} & \frac{1}{2}(\gamma - 1) \\ -\hat{v}c^2 & 0 & c^2 & 0 \\ c^2 - b_1 & (\gamma - 1)\hat{u} & (\gamma - 1)\hat{v} & 1 - \gamma \\ \frac{1}{2}(b_1 - \hat{u}c) & -\frac{1}{2}((\gamma - 1)\hat{u} - c) & -\frac{1}{2}(\gamma - 1)\hat{v} & \frac{1}{2}(\gamma - 1) \end{pmatrix},
\end{aligned} \tag{2.39}$$

with $b_1 = (\gamma - 1)(\hat{u}^2 + \hat{v}^2)/2$. The number of boundary conditions depends on the signs of the eigenvalues $\hat{u}$ and $\hat{u} \pm c$. We also have four cases in the following:

Case 1: $\hat{u} + c < 0$;

Case 2: $\hat{u} < 0, \hat{u} + c \geq 0$;

Case 3: $\hat{u} - c < 0, \hat{u} \geq 0$;

Case 4: $\hat{u} - c \geq 0$.

Note that $\hat{u}$ and c are obtained at the boundary point P_0 in Figure 1. In the 1D case, we consider the above four cases at the left boundary $x = a$, and since we have set up a local coordinate, the $\hat{x}$ -direction is the outward normal direction, then we reverse the order of the four cases. We will briefly elucidate the algorithm in the following, and one can refer to the 1D case for more detailed description of the idea. For convenience, we still take the notations $(w)^{ilw}$ and $(w)^{ext}$ respectively standing for w obtained by the ILW procedure and by extrapolation, and $\mathbf{R} = (r_{ij})_{4 \times 4}$, $\mathbf{R}^{-1} = (\tilde{r}_{ij})_{4 \times 4}$.

For the case 1, all eigenvalues are positive so we have four inflow boundary conditions. We now show the approaches to obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$ and $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ at P_0 . To obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$, we have the following equations.

$$\partial_{\hat{x}}^{(0)} \hat{\mathbf{U}} = (\hat{\mathbf{U}})^{ilw}, \quad \partial_{\hat{x}}^{(l)} \hat{\mathbf{U}} = (\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}})^{ext}, \quad l = 1, \dots, 4. \quad (2.40)$$

After we obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$, then we have the following equations to obtain $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$.

$$\begin{aligned} \partial_{\hat{x}}^{(0)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{F}(\partial_{\hat{x}}^{(0)} \hat{\mathbf{U}}), \quad \partial_{\hat{x}}^{(1)} \mathbf{F}(\hat{\mathbf{U}}) = (\partial_{\hat{x}}^{(1)} \mathbf{F}(\hat{\mathbf{U}}))^{ilw}, \\ \partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}) &= (\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}))^{ext}, \quad l = 2, 3, 4. \end{aligned} \quad (2.41)$$

By the Taylor expansion, we can obtain $\hat{\mathbf{U}}$ and $\mathbf{F}(\hat{\mathbf{U}})$ at P . The case 1 is finished.

For the case 2, we have one positive eigenvalue and three negative eigenvalues, thus we have three boundary conditions. Assume we have the boundary conditions for $\rho, \rho u, \rho v$ at P_0 , then we transform the boundary conditions into $\hat{U}_k(\hat{x}, \hat{y}, t) = g_k(\hat{x}, \hat{y}, t)$, $k = 1, 2, 3$, $(\hat{x}, \hat{y}) \in \partial\Omega$. With the extrapolation and boundary conditions, we can obtain $\hat{\mathbf{U}}$ and matrices $\mathbf{R}$, $\mathbf{R}^{-1}$, $\mathbf{\Lambda}$ at P_0 . We then perform the characteristic decomposition and obtain the characteristic variables $\mathbf{V} = \mathbf{R}^{-1} \hat{\mathbf{U}}$. Since $\hat{u} + c > 0$ and $\hat{u} < 0$, the outgoing characteristic variable is V_4 where $(V_1, V_2, V_3, V_4)^T = \mathbf{V}$. We have the following equations to obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$ at P_0 .

$$\begin{aligned} \partial_{\hat{x}}^{(0)} \hat{U}_k &= g_k(t), \quad k = 1, 2, 3, \quad \partial_{\hat{x}}^{(0)} \hat{U}_4 = \left((V_4)^{ext} - \sum_{k=1}^3 \tilde{r}_{1k} g_k(t) \right) / \tilde{r}_{4k}, \\ \partial_{\hat{x}}^{(l)} \hat{U}_k &= (\partial_{\hat{x}}^{(l)} U_k)^{ext}, \quad k = 1, 2, 3, \quad l = 1, \dots, 4. \end{aligned} \quad (2.42)$$

With Taylor expansion, we obtain $\{\hat{U}_k\}_{k=1}^3$ at P . Then by $\mathbf{V} = \mathbf{R}^{-1}\hat{\mathbf{U}}$ at P we have

$$\hat{U}_4 = \left((V_4)^{ext} - \sum_{k=1}^3 \tilde{r}_{4k} \hat{U}_k \right) / \tilde{r}_{44}. \quad (2.43)$$

To obtain $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ at P_0 , we have the following equations.

$$\begin{aligned} \partial_{\hat{x}}^{(0)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{F}(\partial_{\hat{x}}^{(0)} \hat{\mathbf{U}}), \\ \partial_{\hat{x}}^{(1)} F_k(\hat{\mathbf{U}}) &= (\partial_{\hat{x}}^{(1)} F_k)^{ilw}, \quad k = 1, 2, 3, \\ \partial_{\hat{x}}^{(1)} F_4(\hat{\mathbf{U}}) &= \left((\hat{u} + c)(\partial_{\hat{x}}^{(1)} V_4)^{ext} - \sum_{k=1}^3 \tilde{r}_{4k} \partial_{\hat{x}}^{(1)} F_k(\hat{\mathbf{U}}) \right) / \tilde{r}_{44}, \\ \partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}) &= (\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}))^{ext}, \quad l = 2, 3, 4. \end{aligned} \quad (2.44)$$

By the Taylor expansion, we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P . The case 2 is finished.

For the case 3, we have one inflow boundary condition and three outflow boundary conditions. Assume we have the boundary condition for $\hat{U}_1(\hat{x}, \hat{y}, t) = g_1(\hat{x}, \hat{y}, t)$, $(\hat{x}, \hat{y}) \in \partial\Omega$. With the extrapolation and boundary conditions, we can obtain $\hat{\mathbf{U}}$ and matrices $\mathbf{R}$, $\mathbf{R}^{-1}$, $\mathbf{\Lambda}$ at P_0 . With the characteristic decomposition, we have the characteristic variables $\mathbf{V} = \mathbf{R}^{-1}\hat{\mathbf{U}}$, and V_2, V_3, V_4 are outgoing variables. Then we have the following equations to obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$.

$$\begin{aligned} \partial_{\hat{x}}^{(0)} \hat{\mathbf{U}} &= \mathbf{R} \mathbf{V}, \text{ where } V_k = (V_k)^{ext}, \quad k = 2, 3, 4, \text{ and } V_1 = \left(g_1(t) - \sum_{k=2}^4 r_{1k} (V_k)^{ext} \right) / r_{11}, \\ \partial_{\hat{x}}^{(l)} \hat{U}_1 &= (\partial_{\hat{x}}^{(l)} \hat{U}_1)^{ext}, \quad l = 1, \dots, 4. \end{aligned} \quad (2.45)$$

By the Taylor expansion, we can obtain $\hat{U}_1$ at P , and we then have

$$\hat{\mathbf{U}} = \mathbf{R} \mathbf{V}, \text{ where } V_k = (V_k)^{ext}, \quad k = 2, 3, 4, \text{ and } V_1 = \left(\hat{U}_1 - \sum_{k=2}^4 r_{1k} (V_k)^{ext} \right) / r_{11}. \quad (2.46)$$

Therefore, we obtain $\hat{\mathbf{U}}$ at P . We again emphasize that (2.45) is performed at P_0 while

(2.46) is performed at P . To obtain $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$, we have the equations as follows.

$$\begin{aligned}
\partial_{\hat{x}}^{(0)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{F}(\partial_{\hat{x}}^{(0)} \hat{\mathbf{U}}), \\
\partial_{\hat{x}}^{(1)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{R} \mathbf{\Lambda} \mathbf{V}_{\hat{x}}, \text{ where } \hat{u} \partial_{\hat{x}}^{(1)} V_2 = \hat{u} (\partial_{\hat{x}}^{(1)} V_2)^{ext}, \\
\hat{u} \partial_{\hat{x}}^{(1)} V_3 &= \hat{u} (\partial_{\hat{x}}^{(1)} V_3)^{ext}, (\hat{u} + c) \partial_{\hat{x}}^{(1)} V_4 = (\hat{u} + c) (\partial_{\hat{x}}^{(1)} V_4)^{ext}, \\
\text{and } (\hat{u} - c) \partial_{\hat{x}}^{(1)} V_1 &= ((\partial_{\hat{x}}^{(1)} F_1)^{ilw} - r_{12} \hat{u} \partial_{\hat{x}}^{(1)} V_2 - r_{13} \hat{u} \partial_{\hat{x}}^{(1)} V_3 - r_{14} (\hat{u} + c) \partial_{\hat{x}}^{(1)} V_4) / r_{11}, \\
\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}) &= (\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}))^{ext}, \quad l = 2, 3, 4.
\end{aligned} \tag{2.47}$$

By the Taylor expansion, we obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P . The case 3 is finished.

For the case 4, with the extrapolation we obtain $\hat{\mathbf{U}}$ at P_0 , thus obtain the matrices $\mathbf{R}$, $\mathbf{R}^{-1}$ and $\mathbf{\Lambda}$. Then we perform the characteristic decomposition and obtain the characteristic variables $\mathbf{V} = \mathbf{R}^{-1} \hat{\mathbf{U}}$ and $\{\partial_{\hat{x}}^{(l)} \mathbf{V}\}_{l=0}^4$. With Taylor expansion, we obtain $\mathbf{V}$ at P and then transform it back to $\hat{\mathbf{U}}$ by $\hat{\mathbf{U}} = \mathbf{R} \mathbf{V}$, given by

$$\hat{\mathbf{U}} = \mathbf{R}(\mathbf{V})^{ext}. \tag{2.48}$$

After we obtain $\hat{\mathbf{U}}$ at P , we just plug it into $\mathbf{F}(\hat{\mathbf{U}})$ and obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P . The case 4 is finished.

We summarize the algorithm for 2D problems as follows. Assume we have obtained $\mathbf{U}$ on the interior points at time level t_n , and our goal is to define $\mathbf{U}$ and $\mathbf{F}(\mathbf{U})$ on the ghost point P .

1. For the ghost point P , find the corresponding boundary point $P_0 \in \partial\Omega$. Set up the local coordinate (2.33) and obtain the transformed system (2.34). With the extrapolation and boundary conditions we obtain $\hat{\mathbf{U}}$ at P_0 . Perform the characteristic decomposition at P_0 , and decide the prescribed inflow boundary conditions according to the signs of the eigenvalues of $\mathbf{\Lambda}$ in (2.38).
2. There are four cases of the different signs of eigenvalues of $\mathbf{\Lambda}$ in the following:
 - Case 1: $\hat{u} + c < 0$. We have four inflow boundary conditions at P_0 . We obtain $\{\partial_{\hat{x}}^{(l)} \hat{\mathbf{U}}\}_{l=0}^4$ by (2.40), then we can obtain $\hat{\mathbf{U}}$ at P by the Taylor expansion. Also, we have $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ by (2.41), and we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P by the Taylor expansion.
 - Case 2: $\hat{u} + c \geq 0$, $\hat{u} < 0$. We have three inflow boundary conditions and one outflow boundary condition at P_0 . We firstly obtain $\{\partial_{\hat{x}}^{(l)} \hat{U}_k\}_{l=0}^4$, $k = 1, 2, 3$ by

(2.42), then we can obtain $\{\hat{U}_k\}_{k=1}^3$ at P by the Taylor expansion. With the extrapolation of V_4 , we obtain $\hat{U}_4$ at P by (2.43). Then we have $\{\partial_x^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ by (2.44), and then we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P by the Taylor expansion.

- Case 3: $\hat{u} \geq 0, \hat{u}-c < 0$. We have one inflow boundary condition and three outflow boundary conditions at P_0 . We firstly obtain $\{\partial_x^{(l)} \hat{U}_1\}_{l=0}^4$ by (2.45), then we can obtain $\hat{U}_1$ at P by the Taylor expansion. With the extrapolation of $\{V_k\}_{k=2}^4$, we can obtain $\{\hat{U}_k\}_{k=2}^4$ at P by (2.46). We then have $\{\partial_x^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ by (2.47), and we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ by the Taylor expansion.
 - Case 4: $\hat{u} - c \geq 0$. We have four outflow boundary conditions. By the extrapolation of $\mathbf{V}$, we can obtain $\hat{\mathbf{U}}$ at P by (2.48). And we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ by plugging $\hat{\mathbf{U}}$ into it.
3. After we obtain $\hat{\mathbf{U}}$, we plug it into $\mathbf{G}(\hat{\mathbf{U}})$ and obtain $\mathbf{F}$ and $\mathbf{G}$. With the equations (2.35) and (2.36), we finally obtain $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$.

We remark that for the no-penetration boundary condition $(u, v) \cdot \mathbf{n} = 0$ at solid walls, we would like to use the boundary condition $\rho \hat{u} = 0$ and apply the case 3 in the above algorithm. In this situation, the formula needs some modifications since in the above algorithm we only consider the boundary condition of ρ for illustration purposes. In the following, we show the modifications in the case 3 in the above algorithm for treating the solid wall boundary condition.

Assume we have the boundary condition for $\rho \hat{u} = 0$ at the boundary point P_0 . We obtain $\{\partial_x^{(l)} \hat{U}_1\}_{l=0}^4$ by the following equations.

$$\partial_x^{(0)} \hat{\mathbf{U}} = \mathbf{R} \mathbf{V}, \text{ where } V_k = (V_k)^{ext}, k = 2, 3, 4, \text{ and } V_1 = \left(- \sum_{k=2}^4 r_{2k} (V_k)^{ext} \right) / r_{21}, \quad (2.49)$$

$$\partial_x^{(l)} \hat{U}_2 = (\partial_x^{(l)} \hat{U}_2)^{ext}, \quad l = 1, \dots, 4.$$

Then we can obtain $\hat{U}_2$ at P by the Taylor expansion. With the extrapolation of $\{V_k\}_{k=2}^4$, we can obtain $\{\hat{U}_1, \hat{U}_3, \hat{U}_4\}$ at P by the following equations.

$$\hat{\mathbf{U}} = \mathbf{R} \mathbf{V}, \text{ where } V_k = (V_k)^{ext}, k = 2, 3, 4, \text{ and } V_1 = \left(\hat{U}_2 - \sum_{k=2}^4 r_{2k} (V_k)^{ext} \right) / r_{21}. \quad (2.50)$$

We then obtain $\{\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}})\}_{l=0}^4$ by the following equations.

$$\begin{aligned}
\partial_{\hat{x}}^{(0)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{F}(\partial_{\hat{x}}^{(0)} \hat{\mathbf{U}}), \\
\partial_{\hat{x}}^{(1)} \mathbf{F}(\hat{\mathbf{U}}) &= \mathbf{R} \mathbf{A} \mathbf{V}_{\hat{x}}, \text{ where } \hat{u} \partial_{\hat{x}}^{(1)} V_2 = \hat{u} (\partial_{\hat{x}}^{(1)} V_2)^{ext}, \\
\hat{u} \partial_{\hat{x}}^{(1)} V_3 &= \hat{u} (\partial_{\hat{x}}^{(1)} V_3)^{ext}, (\hat{u} + c) \partial_{\hat{x}}^{(1)} V_4 = (\hat{u} + c) (\partial_{\hat{x}}^{(1)} V_4)^{ext}, \\
\text{and } (\hat{u} - c) \partial_{\hat{x}}^{(1)} V_1 &= (-r_{22} \hat{u} \partial_{\hat{x}}^{(1)} V_2 - r_{23} \hat{u} \partial_{\hat{x}}^{(1)} V_3 - r_{24} (\hat{u} + c) \partial_{\hat{x}}^{(1)} V_4) / r_{21}, \\
\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}) &= (\partial_{\hat{x}}^{(l)} \mathbf{F}(\hat{\mathbf{U}}))^{ext}, \quad l = 2, 3, 4.
\end{aligned} \tag{2.51}$$

Then we can obtain $\mathbf{F}(\hat{\mathbf{U}})$ at P by the Taylor expansion.

2.4 Two-dimensional extrapolation

In this subsection, we consider the two-dimensional extrapolation. Unlike the one-dimensional case, the extrapolation becomes complicated in 2D because the points are usually not well-ordered in the normal direction. Our treatment is to construct the 1D polynomials in the normal direction, then we follow the algorithm of 1D extrapolation to obtain the normal derivatives. To this end, we adopt the least squares method to obtain an interpolating polynomial in 2D, then we can obtain the approximating values along the normal direction and the 1D polynomials are obtained. If the solution is smooth near the boundary, we could just use the high order interpolating polynomial to obtain the normal derivatives and tangential derivatives. However, if there is a discontinuity near the boundary, then we need to do more efforts in constructing the polynomial to make the algorithm more robust. In this subsection, we mainly introduce the two-dimensional WENO type extrapolation.

Assume we have the values $\{f_{ij}\}$ on the grid points of a function $f(x, y)$ in the interior domain. Our goal is to obtain the normal derivatives and tangential derivative, i.e. $\{\frac{\partial^m}{\partial \hat{x}^m} f\}_{m=0}^4$ and $f_{\hat{y}}$ at the boundary point P_0 . Since we may not have the well-ordered points to do the Lagrange extrapolation, we construct the interpolating polynomials by the least squares method. To this end, we just take a stencil $\mathcal{E}$ to obtain the high order approximating polynomial as follows.

$$\mathcal{E} = \left\{ (x_i, y_j) \in \Omega, \sqrt{(x_i - x_{P_0})^2 + (y_j - y_{P_0})^2} \leq R \right\} \tag{2.52}$$

where R is a positive constant, and we take $R = 5h$, $h = \max\{\Delta x, \Delta y\}$ in the numerical tests if not noted otherwise. With the least squares method, we can obtain the 2D polynomial $Q(x, y) \in P^4$ on the stencil $\mathcal{E}$, where $P^k = \text{span}\{x^l y^m, l + m \leq k\}$ is a set of polynomials whose degree of freedom are not greater than k . Thus, near the boundary point P_0 we have

the polynomial $Q(x, y)$ that is a fifth order approximation to $f(x, y)$. When the function $f(x, y)$ is smooth, we can obtain the normal derivatives and tangential derivative at P_0 from the constructed high order polynomial $Q(x, y)$. But when there is a discontinuity near the boundary, special treatment is needed in obtaining $\{\frac{\partial^m}{\partial \hat{x}^m} f\}_{m=0}^4$ and $f_{\hat{y}}$. In the following we describe the two-dimensional WENO type extrapolation based on the multi-resolution WENO method in [32].

First, on the boundary point P_0 we have a segment $\overline{P_0 P_4}$ which is along the normal direction, and $\overline{P_0 P_1} = \cdots = \overline{P_3 P_4} = h$, see Figure 2. Since we have already obtained the high order approximating polynomial $Q(x, y)$, we can obtain the approximating values of f and $f_{\hat{y}}$ on the points $\{P_m\}_{m=0}^4$. Now we have five stencils $S_r = \{P_0, \cdots, P_r\}$, $r = 0, \cdots, 4$ and the approximating values of f and $f_{\hat{y}}$ on them, then we can perform the 1D WENO type extrapolation.

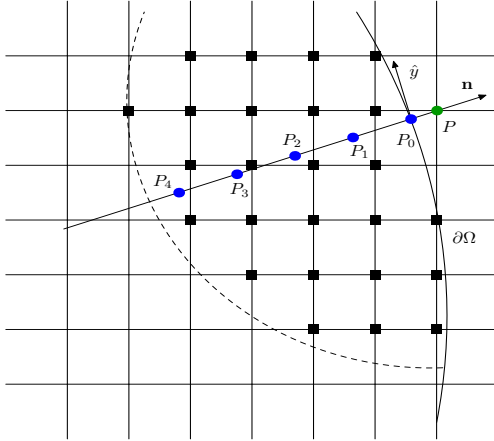


Figure 2: Illustrative sketch for 2D extrapolation: the ghost point P marked in green and its projection point $P_0 \in \partial\Omega$. The interpolating points $P_0, \cdots, P_4$ distributed uniformly along the normal direction and they are marked in blue. The stencils for 2D extrapolation are marked by square symbol in black.

To obtain the normal derivatives $\{\frac{\partial^m}{\partial \hat{x}^m} f\}_{m=0}^4$, we first construct the corresponding 1D polynomials $q_r(\hat{x})$ on S_r , $r = 0, \cdots, 4$. Different from the 1D WENO type extrapolation, we construct the $q_0(\hat{x})$ and $q_r(\hat{x})$, $r = 1, \cdots, 4$ from different 2D interpolating polynomials. Without loss of generality, we assume $\hat{x} = -mh$ at P_m , $m = 0, \cdots, 4$. For the constant approximating polynomial $q_0(\hat{x})$, instead of using the polynomial $Q(x, y)$ to obtain the approximating value on P_0 , we construct a polynomial $Q_0(x, y) \in P^0$ by using the following stencil $\mathcal{E}_0$ with the least squares method.

$$\mathcal{E}_0 = \left\{ (x_i, y_j) \in \Omega, \sqrt{(x_i - x_{P_0})^2 + (y_j - y_{P_0})^2} \leq R_0 \right\},$$

where R_0 is a positive constant, and we take $R_0 = 1.1h$ in the numerical tests if not noted otherwise. Now we take $q_0(\hat{x}) = Q_0(x, y)$. Next, with the approximating polynomial $Q(x, y)$

obtained previously, we then get the approximating values of f on $\{P_m\}_{m=0}^4$. With these approximating values, we can construct the interpolating polynomials $q_r(\hat{x}) \in P^r(\overline{P_0P_4})$ on S_r , $r = 1, \dots, 4$, where $P^k(\overline{P_0P_4})$ is a set of polynomials whose degree is not greater than k on $\overline{P_0P_4}$. So far, we have obtained $q_r(\hat{x})$ on S_r , $r = 0, \dots, 4$, then the 1D WENO type extrapolation can apply and the normal derivatives can be obtained. Here we just briefly write down the procedure of obtaining the normal derivatives $\{\frac{\partial^m}{\partial \hat{x}^m} f\}_{m=0}^4$.

We first present the expressions of $p_r(\hat{x})$ as follows.

$$p_0(\hat{x}) = q_0(\hat{x}), \quad p_r(\hat{x}) = \sum_{m=0}^r \frac{d_m}{d_r} q_r(\hat{x}) - \sum_{m=0}^{r-1} \frac{d_m}{d_r} p_m(\hat{x}), \quad r = 1, \dots, 4, \quad (2.53)$$

where $\{d_r\}_{r=0}^4$ are the linear weights defined in (2.12). Then the nonlinear weights are given as follows.

$$\begin{aligned} \omega_r &= \frac{\alpha_r}{\sum_{s=0}^4 \alpha_s}, \quad \alpha_r = d_r \left(1 + \left(\frac{\tau}{\varepsilon + \beta_r} \right)^4 \right), \quad r = 0, \dots, 4, \\ \tau &= \left(\max_{1 \leq l \leq 3} \{(\beta_l - \beta_4)^2\} \right)^{\frac{1}{2}} + \max_{1 \leq l \leq 4} \{\|q_0(\hat{x}) - q_l(\hat{x})\|^3\} \end{aligned} \quad (2.54)$$

where the $\|\cdot\|$ is the standard L^2 -norm on $(-h, h)$, and the smoothness indicators $\{\beta_r\}_{r=0}^4$ are defined as follows.

$$\begin{aligned} \beta_0 &= c_0 \beta_1, \\ \beta_r &= \sum_{l=1}^r h^{2l-1} \int_{-h}^h \left(\frac{d^l}{d\hat{x}^l} q_r(\hat{x}) \right)^2 d\hat{x}, \quad r = 1, \dots, 4, \end{aligned} \quad (2.55)$$

with $\varepsilon = 10^{-4}$ and c_0 is a positive constant, and we take $c_0 = 0.1$ throughout this paper. Then we have a combination of polynomials $\{p_r(\hat{x})\}_{r=0}^4$.

$$p(\hat{x}) = \sum_{r=0}^4 \omega_r p_r(\hat{x}).$$

With the polynomial $p(\hat{x})$, we have the desired normal derivatives $\{\partial_{\hat{x}}^{(r)} u\}_{r=0}^4$ at P_0 .

$$\partial_{\hat{x}}^{(r)} u = \frac{d^r}{d\hat{x}^r} p(\hat{x}), \quad r = 0, \dots, 4. \quad (2.56)$$

In the algorithm for 2D problems in the previous subsection, we have not written down the explicit formula of the ILW method, but only use the notation $(\cdot)^{ilw}$ instead. In fact, the main difference of the formula between 1D and 2D is that we have the tangential derivatives

in 2D case. If we have the enough boundary conditions, then we could obtain the tangential derivatives with the explicit expressions, but this situation may not always be true. A common way to obtain the tangential derivatives is by extrapolation.

We now proceed to obtain the tangential derivative $\frac{\partial}{\partial \hat{y}}f$. With the 2D approximating polynomial $Q(x, y)$, we can obtain the approximating values of $\frac{\partial}{\partial \hat{y}}f$ on $\{P_m\}_{m=0}^4$. To obtain $\frac{\partial}{\partial \hat{y}}Q(x, y)$, with the chain rule we have

$$Q_{\hat{y}} = Q_x \frac{\partial x}{\partial \hat{y}} + Q_y \frac{\partial y}{\partial \hat{y}} = T_{21}Q_x + T_{22}Q_y. \quad (2.57)$$

where $(T_{ij})_{2 \times 2} = \mathbf{T}$ is the rotation matrix in (2.33). Then we are able to construct the 1D polynomials $\tilde{q}_r(\hat{x})$ on S_r , and $\tilde{q}_r$ is r -th order approximation to $\frac{\partial}{\partial \hat{y}}f$, $r = 1, \dots, 4$. Also, we take $\tilde{q}_0(\hat{x}) = 0$. Note that it would destroy the self-similarity property if we apply the above procedure to $\{\tilde{q}_r(\hat{x})\}_{r=0}^4$ directly. Thus, we adopt the nonlinear weights $\{\omega_r\}_{r=0}^4$ obtained in (2.54), then we have

$$q(\hat{x}) = \sum_{r=0}^4 \omega_r \tilde{q}_r(\hat{x}).$$

where $q(\hat{x})$ is an approximating polynomial of $\frac{\partial}{\partial \hat{y}}f$, Then we obtain the tangential derivative $\frac{\partial}{\partial \hat{y}}f = q(0)$, since we have $\hat{x} = 0$ at P_0 .

3 Numerical tests

In this section, we show the numerical results of one- and two-dimensional problems. We adopt the third order TVD Runge-Kutta method [21] as the time-stepping method. When testing the order of accuracy, to match the order of spatial discretization we take the CFL condition as $\Delta t = O(h^{5/3})$, where h is maximum spatial step size. For the cases containing shocks, we take the CFL condition as $\Delta t = O(h)$. In 1D case, we take the parameter $\delta_1 = 10^{-1}$, $\delta_2 = 10^{-6}$ in (2.2) if not noted otherwise.

3.1 One-dimensional problems

Example 3.1. Consider the one-dimensional linear scalar conservation laws with variable coefficient in the following:

$$\begin{cases} u_t + (\cos(\pi(x+t))u)_x = f_s, & 0 < x < 1, t > 0, \\ u(x, 0) = \sin(\pi x), & 0 < x < 1, \end{cases} \quad (3.1)$$

with the appropriate boundary condition and f_s is the additional source term. The sign of $\cos(\pi(x+t))$ will affect the type of the boundary conditions. In fact, when $\cos(\pi(x+t)) > 0$, then we have inflow boundary condition at $x = 0$, and outflow boundary condition at $x = 1$. And when $\cos(\pi(x+t)) < 0$, we then have inflow boundary condition at $x = 1$, and outflow boundary condition at $x = 0$. If $\cos(\pi(x+t)) = 0$, the equation (3.1) degenerates and we consider it as outflow boundary. We take a suitable source term function f_s so that the exact solution is

$$u(x, t) = \sin(\pi(x - t)).$$

We take the final time $T = 1.2$ to test our algorithm.

From Table 3.1, we can see the designed fifth order convergence in l^1 -, l^2 - and l^∞ -norms.

Table 3.1: Errors and orders of accuracy for solving one-dimensional linear scalar conservation laws (3.1) with final time $T = 1.2$ in Example 3.1.

N	l^1 error	order	l^2 error	order	l^∞ error	order
16	7.71E-04	—	9.66E-04	—	2.76E-03	—
32	1.68E-05	5.52	2.61E-05	5.21	9.90E-05	4.80
64	4.50E-07	5.22	6.51E-07	5.32	2.48E-06	5.32
128	1.34E-08	5.07	1.92E-08	5.08	7.34E-08	5.08
256	4.03E-10	5.06	5.71E-10	5.07	2.18E-09	5.07
512	1.13E-11	5.16	1.56E-11	5.20	6.00E-11	5.18

Example 3.2. Consider the Burgers equation in the following:

$$\begin{cases} u_t + (\frac{u^2}{2})_x = 0, & 0 < x < 3/2, t > 0, \\ u(x, 0) = 1 + 2\sin(\pi x), \end{cases} \quad (3.2)$$

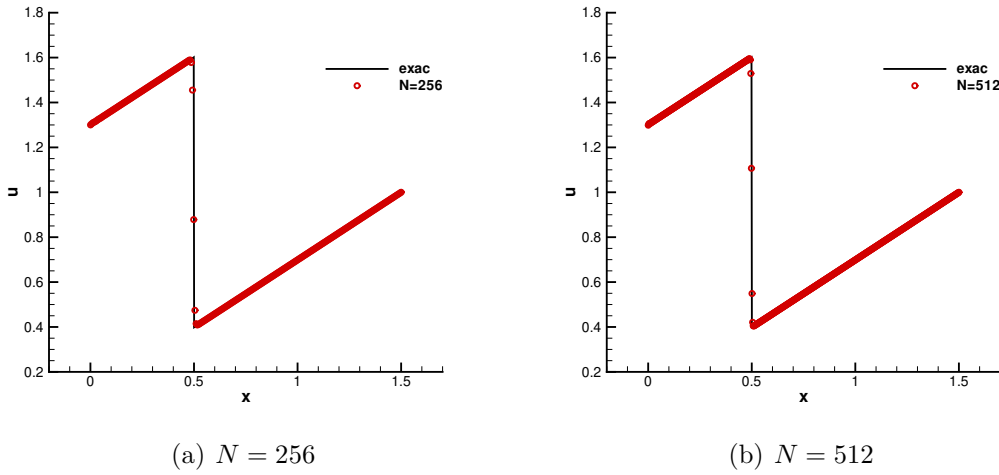
with an appropriate boundary condition. At $x = 0$, when $u > 0$ we have the inflow boundary condition; When $u \leq 0$, we have the outflow boundary condition. At $x = 3/2$, When $u < 0$, we have inflow boundary condition; When $u \geq 0$, we have the outflow boundary condition. We take the final time $T = 0.12$ and $t = 1.5$ to test our algorithm, which represent the smooth case and non-smooth case respectively. Note that u changes its sign at the right boundary before final time $t = 1.5$.

From Table 3.2, we can see the designed fifth order when the exact solution is smooth at time $T = 0.12$. Before time $T = 1.5$, u changes its sign at $x = 3/2$, and we can see the shock is well captured and no instability occurs in Figure 3.

Table 3.2: Errors and orders of accuracy for solving one-dimensional Burgers equation (3.2) with final time $T = 0.12$ in Example 3.2.

N	l^1 error	order	l^2 error	order	l^∞ error	order
16	2.09E-02	—	3.81E-02	—	9.46E-02	—
32	2.93E-03	2.83	8.49E-03	2.16	3.87E-02	1.29
64	3.38E-04	3.12	1.34E-03	2.66	8.11E-03	2.25
128	2.07E-05	4.03	8.84E-05	3.92	5.89E-04	3.78
256	7.27E-07	4.83	3.36E-06	4.72	2.48E-05	4.57
512	2.32E-08	4.97	1.06E-07	4.99	7.95E-07	4.96

Figure 3: Plots of the numerical solution of one-dimensional Burgers equation at $t = 1.5$ in Example 3.2. Solid line: exact solution. Red circles: numerical solution with our boundary treatments.



Example 3.3. Consider the 1D Euler equation (2.16) with an additional source term $\mathbf{f}_s$. With appropriately chosen $\mathbf{f}_s$, we have the exact solution as follows:

$$\rho(x, t) = 1 + 0.2 \sin(x - \sin(\pi t)t), \quad u(x, t) = \sin(\pi t), \quad p(x, t) = 2.$$

The computational domain is $\Omega = (0, 2\pi)$, and we take the final time as $T = 1.4$.

For simplicity, we only show the errors and orders of accuracy for the density ρ in Table 3.3. The number of boundary conditions are determined by the signs of the three eigenvalues of $\mathbf{F}'(\mathbf{U})$, i.e. $\{u - c, u, u + c\}$, $c = \sqrt{\gamma p / \rho}$. At the left boundary $x = 0$, if $u - c > 0$, we have three boundary conditions; If $u - c \leq 0$ and $u > 0$, we have two boundary conditions;

If $u \leq 0$ and $u + c > 0$, we have one boundary condition; If $u + c \leq 0$, then no boundary conditions are imposed. At the right boundary $x = 2\pi$, if $u - c \geq 0$, then we do not have any boundary conditions; If $u \geq 0$ and $u - c < 0$, then we have one boundary condition; If $u + c \geq 0$ and $u < 0$, we have two boundary conditions; If $u + c < 0$, then we have three boundary conditions. In Example 3.3, we only consider the case $u - c < 0, u + c > 0$ and u changes sign at the boundary as time evolves. Note that when $u = 0$, we only have one boundary condition on the boundary. On the boundaries $x = 0$ and $x = 2\pi$, we prescribe the boundary condition for ρ if one boundary condition is required, and for $\rho, \rho u$ if two boundary conditions are required. Table 3.3 show that our algorithm achieves the optimal order of convergence in l^1 -, l^2 - and l^∞ -norms.

Table 3.3: Density errors and orders of accuracy in Example 3.3. On the boundaries, the eigenvalues of $\mathbf{F}'(\mathbf{U})$ are $u - c < 0, u + c > 0$, and u changes sign as time evolves.

N	l^1 error	order	l^2 error	order	l^∞ error	order
16	3.91E-04	—	6.97E-04	—	9.78E-04	—
32	1.98E-05	4.30	2.52E-05	4.30	6.34E-05	3.95
64	7.42E-07	4.74	9.58E-07	4.72	2.19E-06	4.86
128	2.51E-08	4.88	3.36E-08	4.83	1.16E-07	4.23
256	8.01E-10	4.97	1.09E-09	4.95	4.61E-09	4.65
512	2.53E-11	4.98	3.40E-11	5.00	1.63E-10	4.82

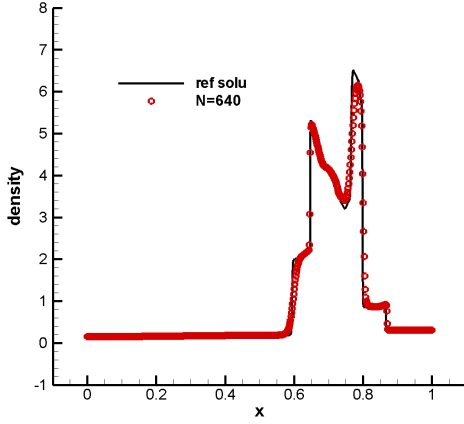
Example 3.4. Consider the interaction of two blast waves problem. The governing equations are 1D Euler equations (2.16) with the initial conditions given as

$$\rho = 1, \quad u = 1, \quad p = \begin{cases} 10^3, & 0 < x < 0.1, \\ 10^{-2}, & 0.1 < x < 0.9, \\ 10^2, & 0.9 < x < 1. \end{cases}$$

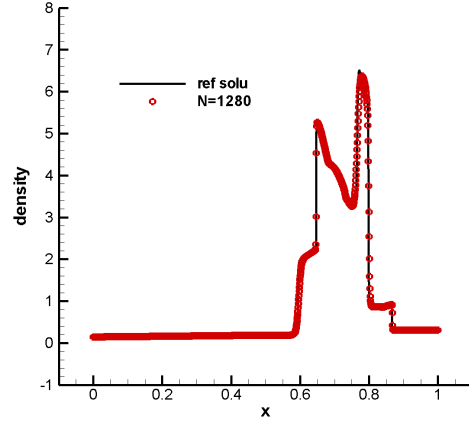
The computational domain is $(0, 1)$, and the final time is $t = 0.038$.

In Figure 4, we show the numerical result of the Example 3.4, indicating that our algorithm works well when considering the solid wall boundary condition. The reference solution is computed on the spatial grid $N = 5120$ with the reflecting technique on the boundaries.

Figure 4: Plots of the density profile at $t = 0.038$ in Example 3.4. Solid line: reference solution is computed by the fifth order WENO scheme with $\Delta x = 1/5120$, together with the reflecting boundary conditions. Red circles: numerical solutions, together with the new ILW boundary treatment.



(a) Numerical solution with $\Delta x = 1/640$



(b) Numerical solution with $\Delta x = 1/1280$

3.2 Two-dimensional problems

Example 3.5. We now test our method for two-dimensional linear scalar hyperbolic conservation laws with variable coefficients in the following:

$$\begin{cases} u_t + \left(\left(\frac{x}{6} + t \right) u \right)_x + 0.2 u_y = f_s & \text{in } \Omega, t > 0, \\ u(x, y, 0) = \sin(x + y) & \text{in } \Omega, \end{cases} \quad (3.3)$$

with $\Omega = (-1, 1) \times (-1, 1)$. We choose suitable boundary conditions and a source function f_s such that the exact solution is

$$u(x, y, t) = \sin(x + y - 0.3t).$$

We divide the domain with the uniform Cartesian mesh as follows.

$$x_i = (i + \delta_1)\Delta x, \quad i = -3, \dots, N_x + 3, \quad y_j = (j + \delta_3)\Delta y, \quad j = -3, \dots, N_y + 3, \quad (3.4)$$

with the mesh step size $\Delta x = 2/(N_x + \delta_1 + \delta_2)$, $\Delta y = 2/(N_y + \delta_3 + \delta_4)$. Then we have $x_0 = -1 + \delta_1\Delta x$, $x_{N_x} = 1 - \delta_2\Delta x$, $y_0 = -1 + \delta_3\Delta y$, $y_{N_y} = 1 - \delta_4\Delta y$. We take $\delta_1 = \delta_3 = 10^{-1}$, $\delta_2 = \delta_4 = 10^{-6}$. The final time is $T = 0.8$.

In Example 3.5, when $x/6 + t > 0$, we have inflow boundary condition at $y = -1$ and outflow boundary condition at $y = 1$. Similarly, When $x/6 + t < 0$, we have outflow boundary condition at $y = -1$ and inflow boundary condition at $y = 1$. When $x/6 + t = 0$ at $y = \pm 1$, we impose the outflow boundary condition. From Table 3.4, we can see that our method achieves the designed fifth order convergence.

Table 3.4: Errors and orders of accuracy in Example 3.5. $\delta_1 = \delta_3 = 10^{-1}$, $\delta_2 = \delta_4 = 10^{-6}$. The final time is $T = 0.8$.

$N_x \times N_y$	l^1 error	order	l^2 error	order	l^∞ error	order
8×10	1.58E-04	—	2.23E-04	—	8.30E-04	—
16×20	3.79E-06	5.38	6.61E-06	5.08	3.66E-05	4.50
32×40	9.93E-08	5.25	1.92E-07	5.11	1.06E-06	5.11
64×80	2.92E-09	5.09	6.29E-09	4.93	3.81E-08	4.80
128×160	8.25E-11	5.15	1.77E-10	5.15	1.17E-09	5.03
256×320	2.35E-12	5.14	5.25E-12	5.08	3.62E-11	5.01

Example 3.6. *We now test our method for two-dimensional linear scalar hyperbolic conservation laws on a disk in the following:*

$$\begin{cases} u_t + u_x + u_y = 0 & \text{in } \Omega, t > 0, \\ u(x, y, 0) = u_0(x, y) & \text{in } \Omega, \end{cases} \quad (3.5)$$

with appropriate boundary conditions. The computational domain Ω is a disk centered at origin with radius 1. We test the following two initial conditions separately:

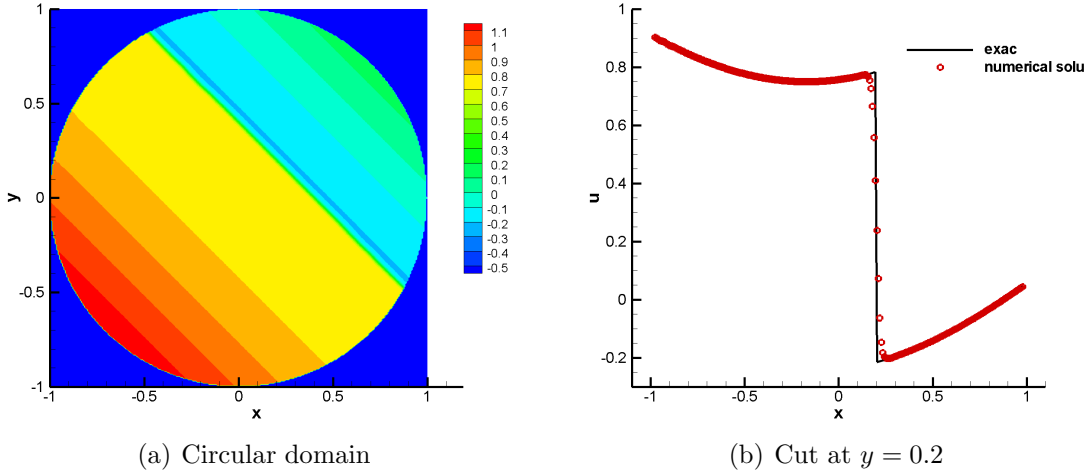
- (a) $u_0(x, y) = \sin(x + y)$,
- (b) $u_0(x, y) = \begin{cases} 0.25 + 0.5 \sin(x + y), & x + y \leq -1.2, \\ 1.25 + 0.5 \sin(x + y), & \text{elsewhere.} \end{cases}$

In Example 3.6, we consider the linear scalar conservation laws (3.5) on a disk and test two kinds of initial boundary conditions. For the initial condition (a), the exact solution is smooth and we have the expected fifth order convergence in Table 3.5. For the initial condition (b), there is a discontinuity in the domain, and we show the contour and cut at $y = 0.2$ of the numerical solution. In [25], it needs a special care when impose the inflow boundary condition on the ghost points near the intersection of the inflow and outflow boundary. By our method, we do not need to treat this case separately and we can see no instability occurring at the boundary.

Table 3.5: Errors and orders of accuracy in Example 3.6 with initial condition (a). The final time is $T = 1.2$.

$N_x \times N_y$	l^1 error	order	l^2 error	order	l^∞ error	order
8×10	1.20E-04	—	2.01E-04	—	7.75E-04	—
16×20	4.98E-06	4.59	9.27E-06	4.44	4.40E-05	4.14
32×40	1.64E-07	4.93	4.39E-07	4.40	4.78E-06	3.20
64×80	5.39E-09	4.92	1.88E-08	4.55	3.18E-07	3.91
128×160	1.56E-10	5.11	7.38E-10	4.67	1.84E-08	4.11
256×320	3.69E-12	5.40	1.64E-11	5.50	6.90E-10	4.74

Figure 5: Plots of the numerical solution in Example 3.6 with initial condition (b). $N_x = 256$, $N_y = 320$. The final time is $T = 0.8$. Left figure: the contour of the numerical solution. Right figure: the cut of the numerical solution at $y = 0.2$. Solid line in black is the exact solution, and the cut of the numerical solution is shown with the red circles.



Example 3.7. We next consider the 2D Burgers equation

$$\begin{cases} u_t + (\frac{u^2}{2})_x + (\frac{u^2}{2})_y = 0 & \text{in } \Omega, t > 0, \\ u(x, y, 0) = 1 + 0.5 \sin(\pi(x + y)) & \text{in } \Omega, \end{cases} \quad (3.6)$$

with appropriate boundary conditions. We consider both the square domain $\Omega = (-1, 1) \times (-1, 1)$ and the circular domain $\Omega = \{(x, y) : x^2 + y^2 \leq 1\}$. For the square domain, the partition is similar as (3.4).

In Example 3.7, we take $R = 4.3h$ in (2.52) for the square domain and $R = 5.5h$ in

(2.52) for the circular domain when performing the 2D extrapolation. We can see the fifth order convergence at least in l^1 - and l^2 -norms in Table 3.6 for both square and circular domains. When we take the final time $T = 1.2$, there is a shock developed in the interior domain, and from Figure 6 we can see the shock is well captured and no instability occurs.

Table 3.6: Errors and orders of accuracy in Example 3.7. The final time is $T = 0.2$.

	$N_x \times N_y$	l^1 error	order	l^2 error	order	l^∞ error	order
Square	8×10	6.51E-04	—	1.15E-03	—	4.94E-03	—
	16×20	3.80E-05	4.10	1.17E-04	3.30	1.52E-03	1.70
	32×40	1.70E-06	4.48	6.01E-06	4.28	9.87E-05	3.94
	64×80	6.94E-08	4.61	3.05E-07	4.30	6.03E-06	4.03
	128×160	2.39E-09	4.74	1.14E-08	4.74	2.54E-07	4.57
	256×320	7.59E-11	5.10	3.71E-10	4.94	9.09E-09	4.81
Disk	8×10	3.47E-04	—	4.93E-04	—	1.46E-03	—
	16×20	2.70E-05	3.69	6.87E-05	2.84	6.79E-04	1.11
	32×40	1.41E-06	4.26	4.03E-06	4.09	4.46E-05	3.93
	64×80	5.85E-08	4.59	1.91E-07	4.40	3.97E-06	3.49
	128×160	2.15E-09	4.86	7.97E-09	4.55	3.85E-07	3.37
	256×320	5.82E-11	5.11	2.50E-10	5.02	1.46E-08	4.72

Example 3.8. Consider the two-dimensional Euler equations (2.32) with appropriate source terms, boundary conditions and initial conditions, such that the exact solutions are given as

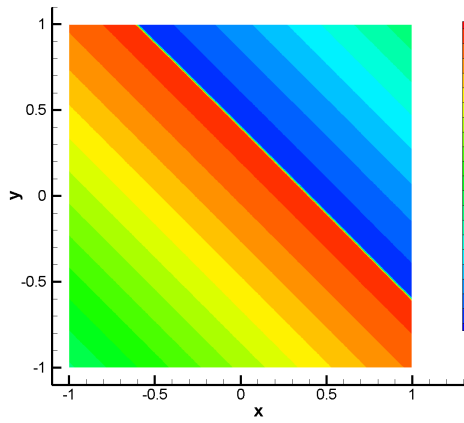
$$\begin{aligned}\rho(x, y, t) &= 1 + 0.2 \sin(x - u(x, y, t)t) \cos(y - v(x, y, t)t), \\ u(x, y, t) &= 0.7 \sin(2\pi t), \quad v(x, y, t) = 0.3 \cos(2\pi t), \quad p(x, y, t) = 1,\end{aligned}$$

The computational domain is $\Omega = (0, 2\pi) \times (0, 2\pi)$, and we take the partition of the domain similar as (3.4). The final time is $T = 0.6$.

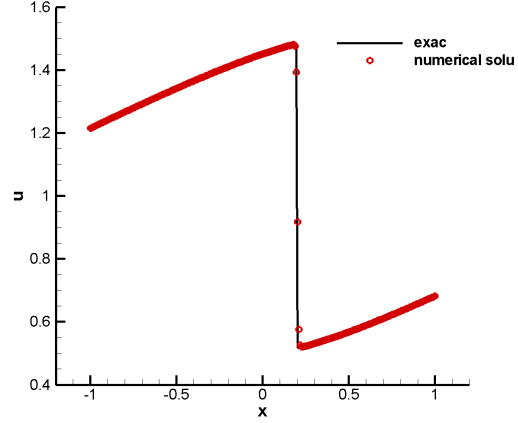
In Example 3.8, u and v change their signs on the boundary as the time evolves. We take the $R = 4.9h$ in 2D extrapolation (2.52). In Table 3.7, we report the density errors and we can see the designed fifth order is achieved at least in l^1 -norm.

Example 3.9. Consider the vortex evolution problem for two-dimensional Euler equation (2.32) (see e.g. [10, 19]). We set the mean flow as $\rho = 1$, $p = 1$, and $(u, v) = (1, 1)$. An isentropic vortex perturbation is added to the mean flow and centered at (x_0, y_0) initially

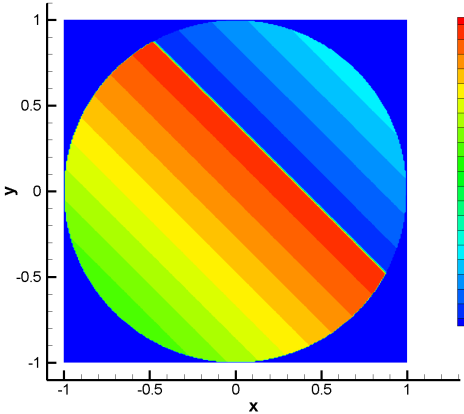
Figure 6: Plots of the numerical solution in Example 3.7. $N_x = 256$, $N_y = 320$. The final time is $T = 1.2$. Top left figure: the contour of the numerical solution on the square domain. Top right figure: the cut of the numerical solution on the square domain at $y = 0.2$. Bottom left figure: the contour of the numerical solution on the disk. Bottom right figure: the cut of the numerical solution on the disk at $y = 0.2$. Solid line in black is the exact solution, and the cut of the numerical solution is shown with the red circles.



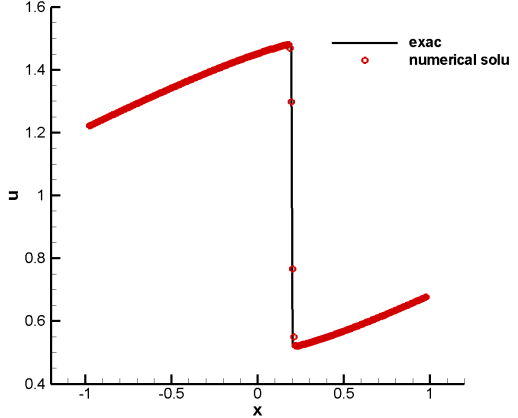
(a) Square domain



(b) The cut at $y = 0.2$



(c) Circular domain



(d) The cut at $y = 0.2$

(perturbation in (u, v) and temperature, no perturbation in the entropy p/ρ^γ):

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)} (-\bar{y}, \bar{x}),$$

$$\delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{(1-r^2)},$$

Table 3.7: Density errors and orders of accuracy in Example 3.8. On the left and right boundaries, the eigenvalues of $\mathbf{F}'(\mathbf{U})$ are $u - c < 0$, $u + c > 0$, and u changes sign as time evolves. On the bottom and upper boundaries, the eigenvalues of $\mathbf{G}'(\mathbf{U})$ are $v - c < 0$, $v + c > 0$, and v changes sign as time evolves.

$N_x \times N_y$	l^1 error	order	l^2 error	order	l^∞ error	order
8×10	3.12E-03	—	3.99E-03	—	1.33E-02	—
16×20	1.64E-04	4.25	2.54E-04	3.98	1.03E-03	3.68
32×40	6.14E-06	4.74	9.17E-06	4.79	4.64E-05	4.48
64×80	1.89E-07	5.02	2.89E-07	4.99	1.62E-06	4.83
128×160	6.20E-09	4.93	1.04E-08	4.80	1.01E-07	4.01
256×320	2.02E-10	4.94	3.66E-10	4.83	3.65E-09	4.79

$$\delta S = 0,$$

where $(\bar{x}, \bar{y}) = (x - x_0, y - y_0)$, $r^2 = \bar{x}^2 + \bar{y}^2$. An simple calculation shows that the exact solution of the vortex evolution problem is that the vortex convected with the mean velocity, and we denote it as $\mathbf{U}_e$. The number of boundary conditions is determined by the signs of four eigenvalues of $\mathbf{F}'(\mathbf{U})$ or $\mathbf{G}'(\mathbf{U})$ on the boundaries, and we take the boundary conditions from $\mathbf{U}_e$ whenever needed. Since the mean flow moves with the velocity $(1, 1)$, the vortex movement is not aligned with the mesh direction. In the computation, we take the vortex strength $\varepsilon = 5$, and $(x_0, y_0) = (0, 0)$. The computational domain is $\Omega = (-0.5, 1) \times (-0.5, 1)$, and the partition of the domain is similar as (3.4). We take the final time is $T = 1$.

In Table 3.8, we report the errors, convergence orders of the density in the vortex evolution in Example 3.9. The eigenvalues change their sign on the boundaries, and we can see the convergence order is around 5 at least in l^1 -norm.

Example 3.10. Consider the double Mach reflection problem [30]. The problem describes a Mach 10 shock horizontally impinges on a ramp inclined by a 30° angle. In order to impose the solid wall boundary condition on the ramp, people usually consider an equivalent problem that a Mach 10 shock initially makes a 60° angle with the horizontal wall and use the reflection technique [10]. With the ILW approach, we are able to solve the original problem with the Cartesian mesh in a single domain. The computational domain is the same as in [25]. We have the initial conditions as follows.

$$(\rho, u, v, p) = \begin{cases} (8, 10, 0, 116.5), & x \leq 0, \\ (1.4, 0, 0, 1), & x > 0. \end{cases}$$

Table 3.8: Density errors and orders of accuracy for the vortex evolution problem in Example 3.9.

$N_x \times N_y$	l^1 error	order	l^2 error	order	l^∞ error	order
8×10	9.84E-04	—	1.42E-03	—	4.95E-03	—
16×20	3.21E-05	4.94	4.26E-05	5.06	1.34E-04	5.21
32×40	1.34E-06	4.58	1.82E-06	4.55	6.00E-06	4.48
64×80	5.12E-08	4.71	6.68E-08	4.77	2.13E-07	4.82
128×160	1.81E-09	4.82	2.46E-09	4.76	1.02E-08	4.38
256×320	6.53E-11	4.80	9.73E-10	4.66	5.90E-10	4.11

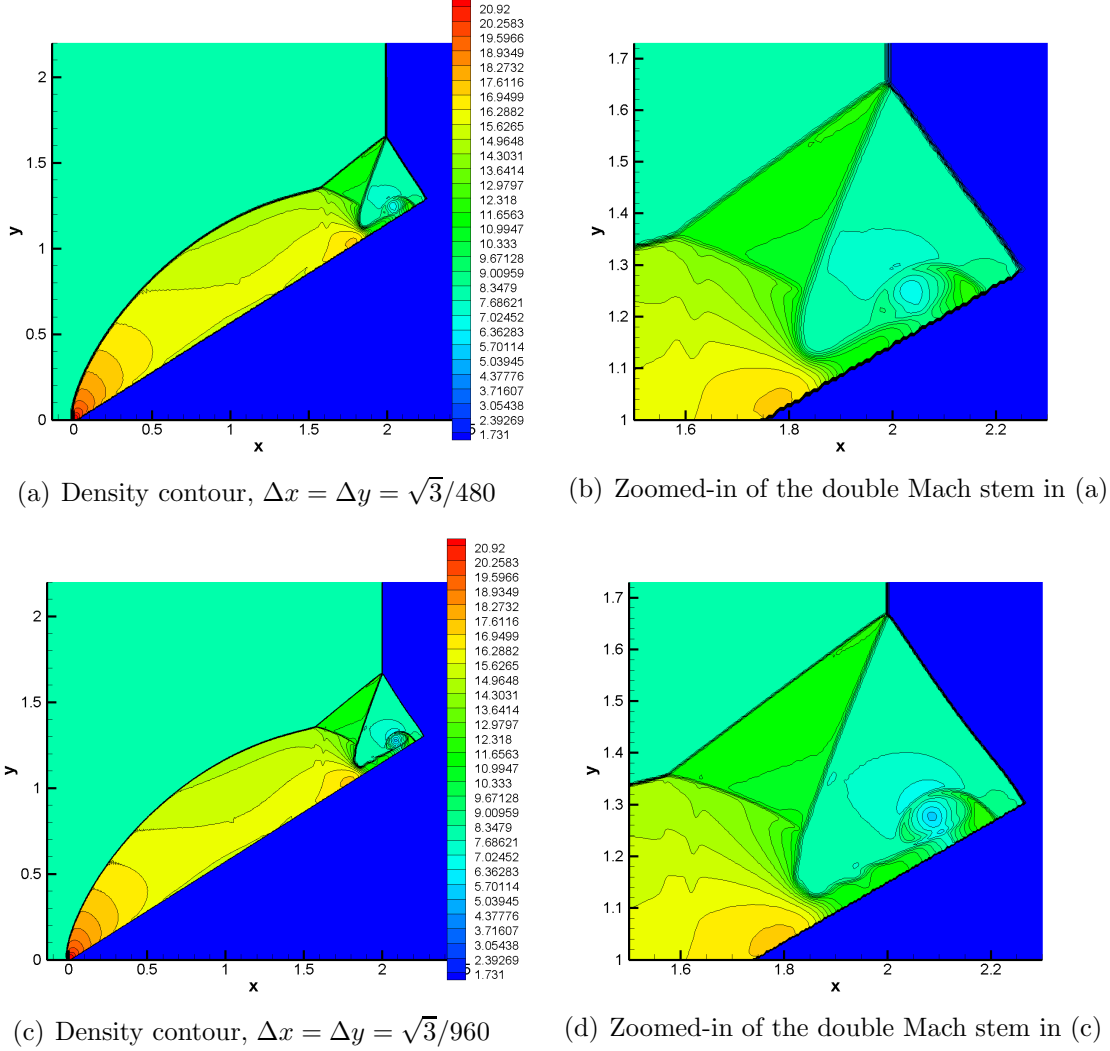
The left and bottom boundary condition is set to be the post-shock condition, and the outflow boundary condition is imposed on the right boundary. On the upper boundary $y = 23/12 + \sqrt{3}/2$, we have the post-shock condition when $x \leq 10t$ and pre-shock condition when $x > 10t$. On the ramp, we use our proposed ILW procedure and the WENO type extrapolation. The final time is taken to be 0.2.

In Figure 7, we show the numerical solution for both $\Delta x = \Delta y = \sqrt{3}/480$ and $\Delta x = \Delta y = \sqrt{3}/960$, and their zoomed-in region near the double Mach stem at time $t = 0.2$ in Example 3.10. It indicates our algorithm works well for treating the solid wall boundary condition.

Example 3.11. Our last example is an inviscid, compressible Mach 3 flow moving towards a circular cylinder from the left. The cylinder locates at the origin with radius 1, and the solid wall boundary condition is imposed on the surface of the cylinder. In [10], a body-fitted curvilinear mesh is used, and it can be transformed to the Cartesian mesh, then the reflection technique can be applied on the surface. With the ILW approach presented in [25], we can solve this problem on the Cartesian grids directly. Different from [25], we take a larger computational domain is $\Omega = (-3, 9) \times (-6, 6)$. At the left boundary $x = -3$, we have supersonic inflow boundary condition, and we have free-stream boundary conditions at the boundaries $x = 9$, $y = \pm 6$. On the surface of the cylinder, the solid wall boundary condition is imposed, i.e. $(u, v) \cdot \mathbf{n} = 0$.

In Example 3.11 we take the final time as $T = 40$, and the numerical solution reaches the steady state in the subregion $(-3, 0) \times (-6, 6)$. In Figure 8, we can see the bow shock is well-captured with the mesh $\Delta x = \Delta y = 1/40$ and $\Delta x = \Delta y = 1/80$.

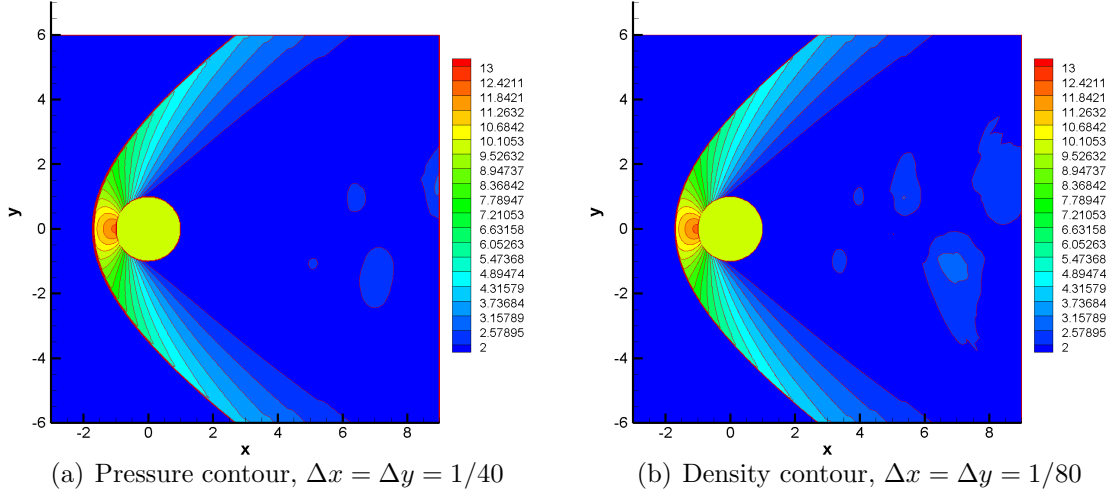
Figure 7: Density contour of double Mach reflection at $t = 0.2$ in Example 3.10. 30 contours from 1.731 to 20.92.



4 Concluding remarks

In this paper we consider a high order boundary treatment for solving hyperbolic conservation laws with high order finite difference methods on a Cartesian mesh. The boundary treatment is very challenging because of the wide stencil of the interior scheme and the domain boundary intersects with the Cartesian mesh in an arbitrary fashion. We propose a new inverse Lax-Wendroff procedure to handle the boundary condition, which could be used for the case when the eigenvalues of the Jacobian matrix are close to zero. Different from [25, 26],

Figure 8: Pressure contour of flow past a cylinder in Example 3.11. 20 contours from 2 to 13.



we perform the inverse Lax-Wendroff procedure on the evaluation of the flux values, thus it avoids the eigenvalues appearing in the denominators. We also propose a new WENO type extrapolation, which will be evoked when there is shock near the boundary. The new WENO type extrapolation can preserve the property of self-similarity, which is desirable in the computation of hyperbolic conservation laws. The idea of the new WENO type extrapolation comes from the multi-resolution WENO schemes [32]. We present extensive numerical examples to validate the good performance of the proposed method, especially for the problems with solid wall boundary condition and the eigenvalues of the Jacobian matrix changing their signs on the boundary. The computational cost of the boundary treatment is not negligible, especially when the extrapolation occupies a large proportion of the boundary treatment in two-dimensional problems. We will consider this issue and try to reduce the computational cost in our future work. Also, we can see that this approach only evaluates the first order derivatives by the inverse Lax-Wendroff procedure, and all other higher order derivatives are obtained by extrapolation. Thus from [13], stability may be an issue when extending this approach to higher order, which will be investigated in our future work. Third, this approach may provide another way to treat the boundary conditions for convection-diffusion problems in [12], since the flux values are evaluated independently.

References

- [1] R. Borges, M. Carmona, B. Costa and W.S. Don, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, J. Comput. Phys. 227 (2008), 3191 – 3211.
- [2] G. Baruch, G. Fibich and S. Tsynkov, *A high-order numerical method for the nonlinear Helmholtz equation in multidimensional layered media*, J. Comput. Phys. 228 (2009), 3789 – 3815.
- [3] M.J. Berger, C. Helzel and R.J. LeVeque, *H-box methods for the approximation of hyperbolic conservation laws on irregular grids*, SIAM J. Numer. Anal. 41 (2003), 893 – 918.
- [4] S. Britt, S. Tsynkov and E. Turkel, *Numerical simulation of time-harmonic waves in inhomogeneous media using compact high order schemes*, Commun. Comput. Phys. (2011), 520 – 541.
- [5] F. Filbet and C. Yang, *An inverse Lax-Wendroff method for boundary conditions applied to Boltzmann type models*, J. Comput. Phys. 245 (2013), 43 – 61.
- [6] B. Gustafsson H.-O. Kreiss and A. Sundström, *Stability theory of difference approximations for mixed initial boundary value problem. II*, Math. Comp. 26 (1972), 649 – 686.
- [7] M. Goldberg and E. Tadmor, *Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. I*, Math. Comp. 32 (1978), 1097 – 1107.
- [8] M. Goldberg and E. Tadmor, *Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. II*, Math. Comp. 36 (1981), 603 – 626.
- [9] L. Huang, C.-W. Shu and M. Zhang, *Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation*, J. Comput. Math. 26 (2008), 336 – 346.
- [10] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. 126 (1996), 202 – 228.

- [11] H.-O. Kreiss, N.A. Petersson and J. Yström, *Difference approximations for the second order wave equation*, SIAM J. Numer. Anal. 40 (2002), 1940 – 1967.
- [12] J. Lu, J. Fang, S. Tan, C.-W. Shu and M. Zhang, *Inverse Lax-Wendroff procedure for numerical boundary conditions of convection-diffusion equations*, J. Comput. Phys. 317 (2016), 276 – 300.
- [13] T. Li, C.-W. Shu and M. Zhang, *Stability analysis of the inverse Lax-Wendroff boundary treatment for high order upwind-biased finite difference schemes*, J. Comput. Appl. Math. 299 (2016), 140 – 158.
- [14] K. Mattsson and M. Almquist, *A high-order accurate embedded boundary method for first order hyperbolic equations*, J. Comput. Phys. 334 (2017), 255 – 279.
- [15] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annu. Rev. Fluid Mech. 37 (2005), 239 – 261.
- [16] M. Medvinsky, S. Tsynkov and E. Turkel, *The method of difference potentials for the Helmholtz equation using compact high order schemes*, J. Sci. Comput. (2012), 150 – 193.
- [17] C.S. Peskin, *The immersed boundary method*, Acta Numer. 11 (2002), 479 – 517.
- [18] V.S. Ryaben’kii, *Method of Difference Potentials and Its Applications*, Springer Series in Computational Mathematics, volume 30, Springer-Verlag, Berlin, 2002.
- [19] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in “Advanced Numerical Approximation of Nonlinear Hyperbolic Equations”, Springer, Berlin, Heidelberg, 1998, 325 – 432.
- [20] C.-W. Shu, *High order WENO and DG methods for time-dependent convection-dominated PDEs: a brief survey of several recent developments*, J. Comput. Phys. 316 (2016), 598 – 613.
- [21] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys. 77 (1988), 439 – 471.
- [22] B. Sjögreen and N. A. Petersson, *A Cartesian embedded boundary method for hyperbolic conservation laws*, Commun. Comput. Phys. 2 (2007), 1199 – 1219.

- [23] I. Singer and E. Turkel, *High-order finite difference methods for the Helmholtz equation*, Comput. Methods Appl. Mech. Engrg. 163 (1998), 343 – 358.
- [24] E. Turkel, D. Gordon, R. Gordon and S. Tsynkov, *Compact 2D and 3D sixth order schemes for the Helmholtz equation with variable wave number*, J. Comput. Phys. (2013), 272 – 287.
- [25] S. Tan and C.-W. Shu, *Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws*, J. Comput. Phys. 229 (2010), 8144 – 8166.
- [26] S. Tan and C.-W. Shu, *A high order moving boundary treatment for compressible inviscid flows*, J. Comput. Phys. 230 (2011), 6023 – 6036.
- [27] 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 Comm. 66, Springer, New York, 2013, 41 – 63.
- [28] S. Tan, C. Wang, C.-W. Shu and J. Ning, *Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws*, J. Comput. Phys. 231 (2012), 2510 – 2527.
- [29] F. Vilar and C.-W. Shu, *Development and stability analysis of the inverse Lax-Wendroff boundary treatment for central compact schemes*, ESAIM: Math. Model. Numer. Anal. 49 (2015), 39 – 67.
- [30] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys. 54 (1984), 115 – 173.
- [31] 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*, J. Sci. Comput. 45 (2010), 514 – 536.
- [32] J. Zhu and C.-W. Shu, *A new type of multi-resolution WENO schemes with increasingly higher order of accuracy*, J. Comput. Phys. 375 (2018), 659 – 683.