Inverse Lax-Wendroff boundary treatment for solving conservation laws with finite difference HWENO methods

Guanghao Zhu[1], Yan Jiang[2] [3], Zhuang Zhao[4] and Mengping Zhang[5]

Abstract: This paper presents a novel inverse Lax-Wendroff (ILW) boundary treatment for finite difference Hermite weighted essentially non-oscillatory (HWENO) schemes to solve hyperbolic conservation laws on arbitrary geometries. The complex geometric domain is divided by a uniform Cartesian grid, resulting in challenge in boundary treatment. The proposed ILW boundary treatment could provide high order approximations of both solution values and spatial derivatives at ghost points outside the computational domain. Distinct from existing ILW approaches, our boundary treatment constructs the extrapolation via optimized through a least squares formulation, coupled with the spatial derivatives at the boundary obtained via the ILW procedure. Theoretical analysis indicates that compared with other ILW methods, our proposed one would require fewer terms by using the relatively complicated ILW procedure and thus improve computational efficiency while preserving accuracy and stability. The effectiveness and robustness of the method are validated through numerical experiments.

Key Words: hyperbolic conservation laws; finite difference HWENO methods; numerical boundary treatment; Inverse Lax-Wendroff method; high order accuracy; stability analysis.

1 Introduction

Hyperbolic conservation laws pose significant challenges in numerical simulations, particularly when complex geometries require accurate and stable boundary treatments.

---

[1]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: zhugy@mail.ustc.edu.cn.

[2]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: jiangy@ustc.edu.cn. Research is supported in part by National Natural Science Foundation of China grant 12271499.

[3]Laoshan Laboratory, Qingdao 266237, China

[4]School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, China. E-mail: zzhao@xmu.edu.cn. Research is supported in part by National Natural Science Foundation of Xiamen, China grant 3502Z202472004 and National Natural Science Foundation of China grant 12401541.

[5]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: mpzhang@ustc.edu.cn.

The high order finite difference Hermite weighted essentially non-oscillatory (HWENO) schemes have emerged as powerful tools for resolving discontinuities and sharp gradients. However, their application to complex geometric domains remains hindered by unresolved computational challenges. In this paper, we propose an efficient boundary treatment for the fifth order finite difference HWENO scheme [5] to solve hyperbolic conservation laws, specifically tailored to address these limitations in irregular domains.

Distinct from traditional weighted essentially non-oscillatory (WENO) schemes, the HWENO scheme introduces derivative equations into its formulation. This strategy utilizes not only the solution values but also their first derivatives, enabling more compact stencils, enhancing resolution, and reducing computational costs at equivalent orders of accuracy compared to the WENO schemes using same nonlinear weight definitions. The HWENO scheme was first proposed by Qiu and Shu [20, 21] in the finite volume framework as a limiter for the discontinuous Galerkin (DG) method. Two high order finite difference HWENO schemes were later developed in [12, 13]. However, due to the presence of mixed derivatives, both finite difference methods reduce to fourth order accuracy in two-dimensional cases. Subsequently, Zhao et al. [29] took the idea of limiter from the DG method to modify the first-order derivatives by applying limiters, enabling the algorithm to achieve fifth-order accuracy in two-dimensional cases. On this basis, the HWENO-L scheme was further proposed in [28], which reduced computational storage and optimized time discretization to improve compactness. Recently, the HWENO-R scheme by Fan et al. [5] represented a significant advancement through optimized reconstruction stencils, demonstrating superior robustness for extreme flow conditions. For more details about the HWENO scheme, please refer to [3, 4, 7, 25, 27]. However, to the best of our knowledge, there is currently no systematic research on the numerical boundary treatments specifically for the HWENO scheme, particularly for complex geometries. This gap motivates our work, which focuses on extending the applicability of the finite difference HWENO scheme to complex geometries through proper numerical boundary treatments.

Within the finite difference framework, two principal challenges arise when implementing HWENO schemes for complex boundaries. First, the HWENO scheme depends on information from neighboring points, which necessitates the construction of values and first derivatives at ghost points near the boundary. Numerical experiments have demonstrated that improper handling of these ghost points can result in loss of accuracy and instability. Secondly, grid points generally do not coincide with physical boundaries, requiring specialized techniques to preserve accuracy while imposing the boundary conditions. Existing boundary treatment strategies broadly fall into two categories: body-fitted mesh methods and non-body-fitted mesh methods. The body-fitted mesh methods ensure that the grid points fall on the boundary, and the numerical solution can accurately satisfy the boundary conditions and achieve high accuracy. However, the mesh

generation requires additional computational cost, particularly for problems involving moving boundaries. The non-body-fitted mesh methods can use uniform Cartesian grids, with the advantages of simple data structure and ease of computation. However, the misalignment between the boundary and grid points often results in the so called "cut-cell" problem, which imposes stricter constraints on time step sizes to ensure numerical stability. To overcome these challenges, a lot of methods have been proposed, including the immersed boundary method [19] and the inverse Lax-Wendroff (ILW) method. It should be noted that the immersed boundary method typically necessitates additional treatment to attain high-order accuracy. In this work, we adopt the ILW method, which naturally achieves high-order accuracy and effectively avoids the cut-cell problem.

The ILW method was proposed by Tan and Shu [22] as a high-order boundary treatment method for solving hyperbolic conservation laws with finite difference WENO schemes, and it has been applied to the simulation of compressible flows in complex domains. This ILW procedure transforms the normal derivatives on the boundary into time derivatives and tangential derivatives utilizing the governing equations repeatedly. With these normal derivatives, the accurate values of ghost points near the boundary are imposed by a Taylor expansion. This method can achieve arbitrary higher-order accuracy at the boundary. To avoid complex calculations caused by the ILW procedure for high-order derivatives, Tan et al. [23] introduced the simplified ILW (SILW), which employs extrapolation to obtain those high-order normal derivatives. A stability analysis for SILW was provided in [10], demonstrating that this method effectively avoids the cut-cell problem and specifying the number of low-order terms required via the ILW procedure. Recently, Liu et al. [16] introduced a new SILW method for the finite different method to further simplify calculations and improve computational efficiency, in which the number of low-order terms required with the ILW procedure is reduced while high-order accuracy and stability are maintained. The SILW method was also extended to the steady state of hyperbolic conservation laws [8], the convection-diffusion equations [9, 11, 18], the Boltzmann equation [6] and the moving boundary problems [2, 14, 15, 17]. All these works are based on the finite difference methods. So far, the ILW boundary treatment has been designed to solve the conservation laws with finite volume method [30] and discontinuous Galerkin method [26] as well.

In this paper, we will propose a novel integration of the ILW method into the fifth order finite different HWENO-R scheme to address high-order ghost point reconstruction for both solution values and derivatives, ensuring stability in irregular domains. This framework will fill a critical methodological void in HWENO applications on complex geometric domains. In particular, a new approximation based on the least square method instead of interpolation is designed in order to use less terms with the complex ILW procedure and reduce computational complexity. Stability analysis will be provided to validate the robustness of the proposed algorithm. Without specified clarification, the

notation "HWENO" in the following refers to the HWENO-R scheme.

The organization of this paper is as follows. In Section 2, we review the framework of the finite difference HWENO scheme. In Section 3, we will describe the ILW method for one-dimensional scalar conservation laws and perform a linear stability analysis using the characteristic spectral visualization method. The algorithm will be extended to two-dimensional systems in Section 4. Section 5 presents numerical experiments to validate the efficiency and high-order accuracy of the proposed algorithm. Finally, conclusions are provided in Section 6.

2 Finite difference HWENO scheme

In this section, we provide a brief review on the finite difference HWENO scheme for hyperbolic conservation laws [5]. We first consider the one-dimensional scalar case:

$$u_t + f(u)_x = 0. \tag{2.1}$$

In the finite difference framework, the grid is uniformly divided into mesh size $\Delta x = x_{i+1} - x_i$. Defining $v = u_x$ and $h(u, v) = f(u)_x = f'(u)u_x = f'(u)v$, we then design the HWENO scheme by adding an additional derivative equation for (2.1):

$$\begin{cases} u_t + f(u)_x = 0, \\ v_t + h(u, v)_x = 0. \end{cases} \tag{2.2}$$

The conservative semi-discrete HWENO scheme can be written as:

$$\begin{cases} \frac{d}{dt}u_i(t) = -\frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}), \\ \frac{d}{dt}v_i(t) = -\frac{1}{\Delta x}(\hat{h}_{i+\frac{1}{2}} - \hat{h}_{i-\frac{1}{2}}). \end{cases} \tag{2.3}$$

Here, $u_i(t)$ and $v_i(t)$ are the approximation of point values $u(x_i, t)$ and $v(x_i, t)$, respectively. $\hat{f}_{i+\frac{1}{2}}$ and $\hat{h}_{i+\frac{1}{2}}$ are the numerical fluxes at half points, which can be obtained via the HWENO reconstruction. To achieve a d-th order accuracy with respect to u, we require

$$\mathcal{F}_i(u, v) := \frac{1}{\Delta x}(\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}) = f(u)_x|_{x_i} + \mathcal{O}(\Delta x^d),$$

$$\mathcal{H}_i(u, v) := \frac{1}{\Delta x}(\hat{h}_{i+\frac{1}{2}} - \hat{h}_{i-\frac{1}{2}}) = h(u, v)_x|_{x_i} + \mathcal{O}(\Delta x^{d-1}).$$

Moreover, to ensure the stability, it is important to use upwinding strategy when constructing the fluxes. One common practice is to use flux splitting. We do a flux splitting on f and h respectively,

$$f_i^\pm = f^\pm(u_i) = \frac{1}{2}(f(u_i) \pm \alpha u_i), \quad h_i^\pm = h^\pm(u_i, v_i) = \frac{1}{2}(h(u_i, v_i) \pm \alpha v_i).$$

where $\alpha = max_u |f'(u)|$. This will ensure that

$$\frac{\partial f^+(u)}{\partial u} \geq 0, \qquad \frac{\partial f^-(u)}{\partial u} \leq 0, \qquad \frac{\partial h^+(u,v)}{\partial v} \geq 0, \qquad \frac{\partial h^-(u,v)}{\partial v} \leq 0.$$

And the upwind-biased reconstruction is used for each of those, meaning that a stencil with one more point from the left (resp. right) will be taken for $\{(f_j^+, h_j^+)\}$ (resp. $\{(f_j^-, h_j^-)\}$) to reconstruct $(\hat{f}_{j+1/2}^+, \hat{h}_{j+1/2}^+)$ (resp. $(\hat{f}_{j+1/2}^-, \hat{h}_{j+1/2}^-)$), further giving us that

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}_{i+\frac{1}{2}}^+ + \hat{f}_{i+\frac{1}{2}}^-, \quad \hat{h}_{i+\frac{1}{2}} = \hat{h}_{i+\frac{1}{2}}^+ + \hat{h}_{i+\frac{1}{2}}^-.$$

In Appendix A, we present the fifth order HWENO construction of $\hat{f}_{i+\frac{1}{2}}^+$ and $\hat{h}_{i+\frac{1}{2}}^+$ given in [5]. The procedures of $\hat{f}_{i+\frac{1}{2}}^-$ and $\hat{h}_{i+\frac{1}{2}}^-$ are similar by applying a mirror symmetry with respect to $x_{i+1/2}$. It is noteworthy that for each grid point x_i, we employ a stencil $S = \{(u_{i+j}, v_{i+j}), j = -2, \cdots, +2\}$.

The algorithm of two-dimensional problems or systems is similar to that of the one-dimensional scalar problem described above. Here, we take the two-dimensional system as an example:

$$\begin{cases} \mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{0}, \\ \mathbf{V}_t + \mathbf{F}^x(\mathbf{U})_x + \mathbf{G}^x(\mathbf{U})_y = \mathbf{0}, \\ \mathbf{W}_t + \mathbf{F}^y(\mathbf{U})_x + \mathbf{G}^y(\mathbf{U})_y = \mathbf{0}, \end{cases} \qquad (2.4)$$

where

$$\mathbf{V} = \mathbf{U}_x, \quad \mathbf{F}^x(\mathbf{U}) = \mathbf{F}'(\mathbf{U})\mathbf{V}, \quad \mathbf{G}^x(\mathbf{U}) = \mathbf{G}'(\mathbf{U})\mathbf{V},$$
$$\mathbf{W} = \mathbf{U}_y, \quad \mathbf{F}^y(\mathbf{U}) = \mathbf{F}'(\mathbf{U})\mathbf{W}, \quad \mathbf{G}^y(\mathbf{U}) = \mathbf{G}'(\mathbf{U})\mathbf{W},$$

and $\mathbf{F}'(\mathbf{U})$ and $\mathbf{G}'(\mathbf{U})$ are the Jacobian matrixes. The semi-discrete finite difference scheme of (2.4) is

$$\begin{cases} \dfrac{d}{dt}\mathbf{U}_{i,j} = -\dfrac{1}{\Delta x}(\hat{\mathbf{F}}_{i+\frac{1}{2},j} - \hat{\mathbf{F}}_{i-\frac{1}{2},j}) - \dfrac{1}{\Delta y}(\hat{\mathbf{G}}_{i,j+\frac{1}{2}} - \hat{\mathbf{G}}_{i,j-\frac{1}{2}}), \\ \dfrac{d}{dt}\mathbf{V}_{i,j} = -\dfrac{1}{\Delta x}(\hat{\mathbf{F}}_{i+\frac{1}{2},j}^x - \hat{\mathbf{F}}_{i-\frac{1}{2},j}^x) - \dfrac{1}{\Delta y}(\hat{\mathbf{G}}_{i,j+\frac{1}{2}}^x - \hat{\mathbf{G}}_{i,j-\frac{1}{2}}^x), \\ \dfrac{d}{dt}\mathbf{W}_{i,j} = -\dfrac{1}{\Delta x}(\hat{\mathbf{F}}_{i+\frac{1}{2},j}^y - \hat{\mathbf{F}}_{i-\frac{1}{2},j}^y) - \dfrac{1}{\Delta y}(\hat{\mathbf{G}}_{i,j+\frac{1}{2}}^y - \hat{\mathbf{G}}_{i,j-\frac{1}{2}}^y). \end{cases} \qquad (2.5)$$

The numerical fluxes $\hat{\mathbf{F}}_{i\pm\frac{1}{2},j}$ and $\hat{\mathbf{F}}_{i\pm\frac{1}{2},j}^x$ are computed using the one-dimensional HWENO reconstruction procedure applied in the x-direction with fixed j. Similarly, the numerical fluxes $\hat{\mathbf{G}}_{i,j\pm\frac{1}{2}}$ and $\hat{\mathbf{G}}_{i,j\pm\frac{1}{2}}^y$ are obtained through the one-dimensional HWENO reconstruction in the y-direction with fixed i. The remaining numerical fluxes, $\hat{\mathbf{F}}_{i\pm\frac{1}{2},j}^y$ and $\hat{\mathbf{G}}_{i,j\pm\frac{1}{2}}^x$, are derived using the fourth-order linear reconstruction. In particular, to mitigate numerical oscillations, the aforementioned reconstruction process is conducted in the characteristic space via characteristic decomposition before being projected back to the physical space.

After the spatial discretization, the third order total variation diminishing (TVD) Runge-Kutta (RK) temporal integration is used for update solution from time t^n to the next level $t^{n+1} = t^n + \Delta t$. For instance, for the 1D HWENO scheme (2.3), we have that

$$
\begin{aligned}
\begin{pmatrix} u_i^{(1)} \\ v_i^{(1)} \end{pmatrix} &= \begin{pmatrix} u_i^n \\ \tilde{v}_i^n \end{pmatrix} - \Delta t \begin{pmatrix} \mathcal{F}_i(u^n, v^n) \\ \mathcal{H}_i(u^n, v^n) \end{pmatrix}, \\
\begin{pmatrix} u_i^{(2)} \\ v_i^{(2)} \end{pmatrix} &= \frac{3}{4} \begin{pmatrix} u_i^n \\ \tilde{v}_i^n \end{pmatrix} + \frac{1}{4} \begin{pmatrix} u_i^{(1)} \\ \tilde{v}_i^{(1)} \end{pmatrix} - \frac{1}{4} \Delta t \begin{pmatrix} \mathcal{F}_i(u^{(1)}, v^{(1)}) \\ \mathcal{H}_i(u^{(1)}, v^{(1)}) \end{pmatrix}, \\
\begin{pmatrix} u_i^{n+1} \\ v_i^{n+1} \end{pmatrix} &= \frac{1}{3} \begin{pmatrix} u_i^n \\ \tilde{v}_i^n \end{pmatrix} + \frac{2}{3} \begin{pmatrix} u_i^{(2)} \\ \tilde{v}_i^{(2)} \end{pmatrix} - \frac{2}{3} \Delta t \begin{pmatrix} \mathcal{F}_i(u^{(2)}, v^{(2)}) \\ \mathcal{H}_i(u^{(2)}, v^{(2)}) \end{pmatrix}.
\end{aligned}
\tag{2.6}
$$

We would like to remark that a nonlinear HWENO correction from v_i to $\tilde{v}_i$ is necessary. This is because the spatial discretization $\mathcal{H}_i$ is a linear scheme, which can induce numerical oscillations near discontinuities. For two-dimensional cases, the HWENO correction is also added at each time step to modify the derivatives in a dimension-by-dimension manner. The specific procedure for this correction in 1D is detailed in Appendix B.

It is noted in [1] that special care must be taken when we impose the time dependent boundary conditions. Assuming we have the time dependent Dirichlet boundary condition $g(t)$, the boundary conditions for the intermediate stages of the RK scheme (2.6) must be carefully adjusted to maintain the desired third order of accuracy,

$$
\begin{aligned}
u^n &\sim g(t_n), \\
u^{(1)} &\sim g(t_n) + \Delta t g'(t_n), \\
u^{(2)} &\sim g(t_n) + \frac{1}{2} \Delta t g'(t_n) + \frac{1}{4} \Delta t^2 g''(t_n).
\end{aligned}
\tag{2.7}
$$

3 1D ILW boundary treatment

At first, we illustrate the idea of the inverse Lax-Wendroff procedure for HWENO reconstruction of 1D scalar conservation laws

$$
\begin{cases}
u_t + f(u)_x = 0, & x \in (0,1),\ t > 0, \\
u(0,t) = g(t), & t > 0, \\
u(x,0) = u_0(t), & x \in [0,1].
\end{cases}
\tag{3.1}
$$

with the assumption $f'(u) > 0$. This assumption guarantees the left boundary $x = 0$ is an inflow boundary where a boundary condition should be imposed and the right boundary $x = 1$ is an outflow boundary where no boundary condition is needed.

Suppose that the grid is divided uniformly as

$$
0 \leq C_a \Delta x = x_1 < x_2 < ... < x_N = 1 - C_b \Delta x \leq 1,
\tag{3.2}
$$

with $C_a, C_b \in [0, 1)$. We construct the HWENO scheme (2.3) on each inner grid point, $i = 1, \ldots, N$. Due to the wide stencil used in HWENO procedure, we need to construct the ghost points values $\{(u_i, v_i)\}$ on $x_{1-i} = x_1 - i\Delta x$ and $x_{N+i} = x_N + i\Delta x$, $i = 1, 2$, with sufficient accuracy. For the outflow boundary condition, we can get the ghost point values via extrapolation directly, or HWENO-type extrapolation to avoid oscillations and maintain accuracy, which will be given later. Meanwhile, for the inflow boundary condition, we will get them via the ILW boundary treatment in which the given boundary condition is taken into account.

As the ILW procedure outlined in previous work [22, 23] operates independently of the interior scheme, we can easily apply it in HWENO scheme directly. The fundamental idea of the ILW procedure is transforming the spatial derivatives at the boundary into time derivatives iteratively utilizing the equations and boundary conditions. Therefore, for the 1D problem (3.1), the ILW procedure will provide us with

$$
\begin{aligned}
&\partial_x^{(0)} u|_{x=0} = g(t), \\
&\partial_x^{(1)} u|_{x=0} = -\frac{g'(t)}{f'(g(t))}, \\
&\partial_x^{(2)} u|_{x=0} = \frac{f'(g(t))g''(t) - 2f''(g(t))g'(t)^2}{f'(g(t))^3}, \\
&\ldots
\end{aligned}
\tag{3.3}
$$

Then we can construct the d-th order approximation polynomial through Taylor expansion

$$
\tilde{u}(x) = \sum_{k=0}^{d-1} \frac{x^k}{k!} \partial_x^{(k)} u|_{x=0},
\tag{3.4}
$$

and define

$$
u_j = \tilde{u}(x_j) = \sum_{k=0}^{d-1} \frac{(x_j)^k}{k!} \partial_x^{(k)} u|_{x=0}, \quad v_j = \tilde{u}'(x_j) = \sum_{k=1}^{d-1} \frac{(x_j)^{k-1}}{(k-1)!} \partial_x^{(k)} u|_{x=0}, \quad j = 0, -1.
\tag{3.5}
$$

This idea comes from the traditional ILW method for hyperbolic conservation laws proposed by Tan and Shu [22] and will guarantee the global d-th order accuracy. However, the cost of deriving higher order spatial derivatives becomes prohibitive, especially for multi-dimensional problems and systems. To address this, the simplified ILW (SILW) method [23] was proposed when coupling with finite difference schemes, in which the extrapolation is allowed for some higher order derivatives to avoid complex calculations. On a stable premise, for a d-th order boundary treatment, the ILW process is used to get the spatial derivatives at boundary below order $k_d - 1$, while extrapolation is applied for those derivatives of order k_d and above. The relation between d and k_d is studied

in [10]. Recently, Liu et al. [16] further reduced k_d by introducing artificial points, thereby improving computational efficiency.

In fact, we can apply those ideas to construct the ghost point values and derivatives for the HWENO scheme. However, the stability analysis conclusion is suboptimal. For example, for $d = 5$, the traditional SILW method requires $k_d = 4$ for the fifth-order HWENO scheme, while Liu's modified SILW method requires $k_d = 3$. In this paper, we improve the Liu's work, allowing the scheme to achieve overall stability with only $k_d = 2$.
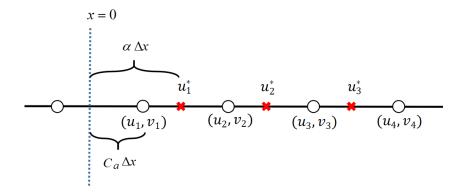
3.1 Fifth order SILW boundary treatment.

Figure 3.1. Fifth order SILW boundary treatment diagram.

Algorithm 1. Fifth order SILW method for 1D scalar cases.

Step 1. Use least square method to construct a quartic polynomial $p(x)$ with inner grid point values $\{(u_i, v_i), 1 \le i \le k\}$,

$$\min_{p(x) \in P^4} \sum_{i=1}^{k} \left(\|p(x_i) - u_i\|^2 + \Delta x^2 \|p'(x_i) - v_i\|^2 \right).$$

To ensure the existence of $p(x)$, we require $k \ge 3$.

Step 2. We use $p(x)$ to construct some special point values

$$u_i^* = p(i \, \alpha \, \Delta x), \quad 1 \le i \le 5 - k_d.$$

The parameter $\alpha > 0$ will be given later based on the stability analysis.

Step 3. Utilizing the boundary conditions and the ILW procedure to get the spatial derivatives at boundary $\partial_x^{(i)} u|_{x=0}$, $i \le k_d - 1$.

8

Step 4. Construct a quartic polynomial $q(x)$ satisfying

$$\begin{cases} \dfrac{d^{(i)}}{dx^i} q(0) = \partial_x^{(i)} u|_{x=0}, & 0 \le i \le k_d - 1, \\ q(i\,\alpha\,\Delta x) = u_i^*, & 1 \le i \le 5 - k_d. \end{cases}$$

Step 5. Obtain the values and derivatives at ghost points

$$u_i = q(x_i), \quad v_i = q'(x_i), \quad i = 0, -1.$$

Compared with Liu's framework, the main difference is that we introduce an additional parameter k and incorporate the least squares approach rather than interpolation in Step 1. As the subsequent stability analysis will demonstrate, our method could employ smaller k_d, exhibiting superior stability and efficiency. In the algorithm mentioned above, there are three parameters: k, k_d and α. Both k and k_d should be as small as possible for better computational efficiency. However, too small parameters may affect the stability of the algorithm. Therefore, we will later determine the selection of k, k_d and α through stability analysis.

3.2 Fifth order HWENO-type extrapolation

When the shock wave approaches the boundary, higher order extrapolation may lead to numerical oscillations. To prevent this from happening, we need to design a robust WENO-type extrapolation in Step 4 of Algorithm 1. The fifth order WENO-type extrapolation based on $\{u|_{x=0}, \partial_x u|_{x=0}, u_1^*, u_2^*, u_3^*\}$ in Algorithm 1 will be described in detail.

Algorithm 2. Fifth order HWENO-type extrapolation in SILW process.

Step 1. Construct five polynomials $p_0(x), p_1(x), p_2(x), p_3(x), p_4(x)$ of orders ranging from 0 to 4, satisfying

$$\begin{aligned} p_0(0) &= u|_{x=0}, \\ p_1(0) &= u|_{x=0}, \quad p_1'(0) = \partial_x u|_{x=0}, \\ p_2(0) &= u|_{x=0}, \quad p_2'(0) = \partial_x u|_{x=0}, \quad p_2(i\,\alpha\Delta x) = u_i^*, \ i = 1, \\ p_3(0) &= u|_{x=0}, \quad p_3'(0) = \partial_x u|_{x=0}, \quad p_3(i\,\alpha\Delta x) = u_i^*, \ i = 1, 2, \\ p_4(0) &= u|_{x=0}, \quad p_4'(0) = \partial_x u|_{x=0}, \quad p_4(i\,\alpha\Delta x) = u_i^*, \ i = 1, 2, 3. \end{aligned}$$

And define their linear weights d_r to be

$$d_0 = \Delta x^4, \ d_1 = \Delta x^3, \ d_2 = \Delta x^2, \ d_3 = \Delta x, \ d_4 = 1 - \sum_{i=0}^{3} d_i.$$

Step 2. Define the smoothness indicators

$$\beta_i = \sum_{j=1}^{i} \int_{-\Delta x/2}^{\Delta x/2} \Delta x^{2j-1} (\frac{d^j p_i(x)}{dx^j})^2 dx, \quad i = 1, 2, 3, 4, \tag{3.6}$$

which is used to measure the smoothness of each polynomial. To ensure β_i at the same magnitude in the smooth case, we take $\beta_0 = \Delta x^2$.

Step 3. Next, express the nonlinear weight ω_i as

$$\omega_i = \frac{\gamma_i}{\sum_{j=0}^{4} \gamma_j}, \quad \gamma_i = \frac{d_i}{(\epsilon + \beta_i)^2}.$$

Here, ϵ is used to avoid the denominator being zero, and it is typically set to $\epsilon = 10^{-6}$.

Step 4. Finally, we can ultimately obtain the extrapolation polynomial $p(x)$

$$p(x) = \sum_{i=0}^{4} \omega_i p_i(x).$$

With the help of smoothness indicator and nonlinear weights, the extrapolation can achieve fifth order accuracy when the solution is smooth near boundary, and degenerate to lower order scheme to avoid spurious oscillation if there is a discontinuity close to the boundary.

For the least squares method, we can use a similar HWENO-type extrapolation. A constant polynomial $p_0(x)$ is constructed from $\{u_0\}$. A quadratic polynomial $p_1(x)$ is obtained by least squares fitting $\{u_1, v_1, u_2, v_2\}$, and a quartic polynomial $p_2(x)$ is obtained by least squares fitting $\{u_1, v_1, u_2, v_2, u_3, v_3\}$. The linear weights are $(d_0, d_1, d_2) = (\Delta x^4, \Delta x^2, 1 - \Delta x^2 - \Delta x^4)$. The nonlinear weights $\omega_0, \omega_1, \omega_2$ are computed as above, giving the extrapolation polynomial $p(x) = \omega_0 p_0(x) + \omega_1 p_1(x) + \omega_2 p_2(x)$.

On the other hand, the HWENO-type extrapolation can be applied for the outflow boundary $x = b$ as well. We construct a linear polynomial $p_0(x)$ using the interior points $\{u_N, v_N\}$, a cubic polynomial $p_1(x)$ using $\{u_N, v_N, u_{N-1}, v_{N-1}\}$, and a fifth-degree polynomial $p_2(x)$ using $\{u_N, v_N, u_{N-1}, v_{N-1}, u_{N-2}, v_{N-2}\}$. The linear weights are assigned as $(d_0, d_1, d_2) = (\Delta x^4, \Delta x^2, 1 - \Delta x^2 - \Delta x^4)$. Finally, the nonlinear weights $\omega_0, \omega_1, \omega_2$ are constructed using the same method as described above, yielding the extrapolation polynomial $p(x) = \omega_0 p_0(x) + \omega_1 p_1(x) + \omega_2 p_2(x)$. Thus, we can obtain $u_{N+i} = p(x_{N+i})$ and $v_{N+i} = p'(x_{N+i})$, $i = 1, 2$.

3.3 Stability analysis

Based on the advection equation $u_t + cu_x = 0$, we analyze the stability of the fully discrete scheme, employing the fifth order linear upwind scheme, third order TVD Runge-Kutta time evolution and the proposed boundary treatment. Assume $c > 0$, thus the upwind linear scheme can be expressed as

$$\frac{du_i}{dt} = -\frac{c}{\Delta x}\left(\frac{23}{120}u_{i-2} - \frac{33}{40}u_{i-1} + \frac{3}{40}u_i + \frac{67}{120}u_{i+1} + \frac{3\Delta x}{40}v_{i-2} - \frac{3\Delta x}{40}v_{i-1} + \frac{7\Delta x}{40}v_i - \frac{7\Delta x}{40}v_{i+1}\right),$$

$$\frac{dv_i}{dt} = -\frac{c}{\Delta x}\left(-\frac{3}{8\Delta x}u_{i-2} + \frac{19}{8\Delta x}u_{i-1} - \frac{29}{8\Delta x}u_i + \frac{13}{8\Delta x}u_{i+1} - \frac{1}{8}v_{i-2} + \frac{1}{8}v_{i-1} + \frac{3}{8}v_i - \frac{3}{8}v_{i+1}\right).$$

$$(3.7)$$

Let us start with the periodic boundary problems. Substituting $u(x,t) = \frac{1}{\sqrt{2\pi}}e^{i\omega x}\hat{u}(\omega, t)$ and $v(x,t) = \frac{1}{\sqrt{2\pi}}e^{i\omega x}\hat{v}(\omega, t)$ into the above equations gives us a matrix-vector form

$$\frac{d\hat{\mathbf{u}}}{dt} = \frac{c}{\Delta x}\mathcal{Q}_1\hat{\mathbf{u}}. \tag{3.8}$$

Here $\hat{\mathbf{u}} = (\hat{u}, \hat{v})^T$. The matrix $\mathcal{Q}_1$ can be expanded as

$$\mathcal{Q}_1 = \begin{pmatrix} q_1 & q_2 \\ q_3 & q_4 \end{pmatrix},$$

with $\xi = \omega\Delta x$ and

$$\begin{cases} q_1 = \left(-\frac{23}{120}\cos 2\xi + \frac{4}{15}\cos\xi - \frac{3}{40}\right) + \left(\frac{23}{120}\sin 2\xi - \frac{83}{60}\sin\xi\right)i, \\ q_2 = \Delta x\left[\left(-\frac{3}{40}\cos 2\xi + \frac{1}{4}\cos\xi - \frac{7}{40}\right) + \left(\frac{3}{40}\sin 2\xi + \frac{1}{10}\sin\xi\right)i\right], \\ q_3 = \frac{1}{\Delta x}\left[\left(\frac{3}{8}\cos 2\xi - 4\cos\xi + \frac{29}{8}\right) + \left(-\frac{3}{8}\sin 2\xi + \frac{3}{4}\sin\xi\right)i\right], \\ q_4 = \left(\frac{1}{8}\cos 2\xi + \frac{1}{4}\cos\xi - \frac{3}{8}\right) + \left(-\frac{1}{8}\sin 2\xi + \frac{1}{2}\sin\xi\right)i. \end{cases}$$

It is noted that the HWENO scheme uses the HWENO limiter to modify the first-order derivative v_i in each RK stage to suppress oscillations. If we consider the linear modification, then $\tilde{v}_i$ can be expressed as

$$\tilde{v}_i = \frac{3}{4\Delta x}(u_{i+1} - u_{i-1}) - \frac{1}{4}(v_{i+1} + v_{i-1}). \tag{3.9}$$

The corresponding modification matrix form is

$$\begin{pmatrix} \hat{u} \\ \hat{\tilde{v}} \end{pmatrix} = \mathcal{Q}_2\begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}, \qquad \mathcal{Q}_2 = \begin{pmatrix} 1 & 0 \\ \frac{3}{2\Delta x}\sin\xi i & -\frac{1}{2}\cos\xi \end{pmatrix}.$$

Combined with the third order TVD RK time evolution (2.6) and denote $\lambda_{cfl} = \frac{c\Delta t}{\Delta x}$, we have that

$$\hat{\mathbf{u}}^{n+1} = \mathcal{G}\hat{\mathbf{u}}^n,$$

and the matrix $\mathcal{G}$ can be written as follows

$$\mathcal{G} = \frac{1}{3}\mathcal{Q}_2 + \frac{2}{3}(\mathcal{Q}_2 + \lambda_{cfl}\mathcal{Q}_1)(\frac{3}{4}\mathcal{Q}_2 + \frac{1}{4}(\mathcal{Q}_2 + \lambda_{cfl}\mathcal{Q}_1)(\mathcal{Q}_2 + \lambda_{cfl}\mathcal{Q}_1)). \qquad (3.10)$$

In order to ensure the stability of the fully discrete scheme, the two eigenvalues of the matrix $\mathcal{G}$ (z_1 and z_2) have to fall inside the unit circle, i.e., $|z_1| \leq 1, |z_2| \leq 1$ for all ξ. It is worth noting that the Δx and $\frac{1}{\Delta x}$ will cancel each other out when calculating the eigenvalues, resulting that z_1 and z_2 are independent of Δx. It is easy to prove that the stability condition is $\lambda_{cfl} \leq 1.07$. We plot the absolute value of the maximum eigenvalue $|z_G| = \max(|z_1|, |z_2|)$ in Figure 3.2 to verify the CFL condition.
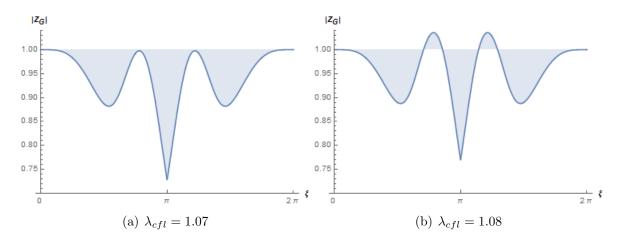

(a) $\lambda_{cfl} = 1.07$ (b) $\lambda_{cfl} = 1.08$

Figure 3.2. The absolute value of the maximum eigenvalue $|z_G|$ with respect to ξ.

Next, we will concentrate on our methods for solving the linear advection equation involving homogeneous Dirichlet conditions $g(t) = 0$ at $x = 0$. Our starting point is to ensure that the boundary treatment will not affect the stability of inner scheme, that is the scheme can maintain stable under $\lambda_{cfl} \leq 1.07$ for any $C_a \in [0, 1)$.

Initially, we express the semi-discrete scheme (3.7) with the proposed numerical boundary treatments in matrix-vector form

$$\frac{d}{dt}\mathbf{U} = \frac{c}{\Delta x}\mathbf{Q}_1\mathbf{U}, \qquad (3.11)$$

where, $\mathbf{U} = (u_1, \cdots, u_N, v_1, \cdots, v_N)^T$, and $\mathbf{Q}_1$ is the coefficient matrix corresponding to the spatial discretization and boundary treatment. At the same time, we can obtain the modification matrix of the corresponding spatial discretization (3.9)

$$\tilde{\mathbf{U}} = \mathbf{Q}_2\mathbf{U}. \qquad (3.12)$$

Combined with the time evolution (2.6), the fully discrete can be presented as

$$\mathbf{U}^{n+1} = \mathbf{G}\mathbf{U}^n, \tag{3.13}$$

with the matrix

$$\mathbf{G} = \frac{1}{3}\mathbf{Q}_2 + \frac{2}{3}(\mathbf{Q}_2 + \lambda_{cfl}\mathbf{Q}_1)(\frac{3}{4}\mathbf{Q}_2 + \frac{1}{4}(\mathbf{Q}_2 + \lambda_{cfl}\mathbf{Q}_1)(\mathbf{Q}_2 + \lambda_{cfl}\mathbf{Q}_1)). \tag{3.14}$$

Relevant studies [10, 24] indicate that to ensure the fully discrete scheme is stable, we only need to focus on the fixed eigenvalues of the matrix $\mathbf{G}$ that do not change with respect to the number of grids N, i.e., $|z_{\mathbf{G}}| \leq 1$ holds for $\lambda_{cfl} \leq 1.07$ and all $C_a \in [0, 1)$. Furthermore, it would be even better if the algorithm is stable with k_d and k as less as possible, as this not only ensures a reduced computational cost, but also matches the compactness of HWENO scheme.

Table 3.1 shows the range of parameter α in our algorithm under the stability conditions for different k and k_d. For example, when $k = 3$ and $k_d = 1$, no parameter α exists to ensures the the algorithm stable for all $C_a \in [0, 1)$. When $k = 3$ and $k_d = 2$, the algorithm is stable if α is chosen within the interval $[0.93, 1.09]$. Therefore, in the one-dimensional cases, we select $k = 3$ and $k_d = 2$ to implement our algorithm.

Table 3.1. Linear stability analysis results: reasonable range of parameter α.

k_d \\ k	3	4
$k_d=1$	-	-
$k_d=2$	[0.93,1.09]	[0.84,1.74]
$k_d=3$	[0.77,10+]	[0.78,10+]

4   2D ILW boundary treatment

Consider the two-dimensional compressible Euler equations along with suitable boundary and initial conditions. The governing equation is given as

$$\mathbf{U}_t + \mathbf{F}(\mathbf{U})_x + \mathbf{G}(\mathbf{U})_y = \mathbf{0}, \quad (x, y) \in \Omega, \tag{4.1}$$

with

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(E+p) \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}, \tag{4.2}$$

and

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2).$$

Here, ρ is the density, $\mathbf{u} = (u, v)^T$ is the velocity, p is the pressure, E is the total energy, and $\gamma = 1.4$ for ideal polytropic gas.

The domain Ω can be arbitrary complex geometry. We assume the domain is covered by a uniform non-body-fitting Cartesian mesh, with

$$\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}, \quad \Delta y = y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}, \quad h = \sqrt{\Delta x^2 + \Delta y^2}.$$

Suppose Δx and Δy are of the same magnitude.

Since the HWENO method requires information from neighboring points, we need to construct information for ghost points that fall outside the domain. Assuming that $P_{i,j}$ is a ghost point. For the fifth order HWENO scheme, we need to construct the point value $\mathbf{U}_{i,j}$ with fifth order accurate and the first derivatives $(\mathbf{U}_x)_{i,j}, (\mathbf{U}_y)_{i,j}$ with fourth order accurate via SILW method.
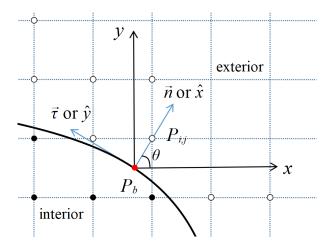

Figure 4.3. Two dimensional ILW method diagram

Similar to [16,22,23], at first, we need to perform a local coordinate transformation to convert the 2D extrapolation into 1D cases. Find the corresponding foot point $P_b \in \partial\Omega$ whose outward normal vector passes through $P_{i,j}$, and denote the unit outward normal vector as $\mathbf{n} = (\cos\theta, \sin\theta)$, see Figure 4.3. At point P_b, we set up a local coordinate system such that the $\hat{x}$-axis aligns with the direction of $\mathbf{n}$ and the $\hat{y}$-axis is in the tangential direction of $\partial\Omega$, i.e.,

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \tag{4.3}$$

In the new coordinate system, the equation (4.1) can be expressed as

$$\hat{\mathbf{U}}_t + \mathbf{F}(\hat{\mathbf{U}})_{\hat{x}} + \mathbf{G}(\hat{\mathbf{U}})_{\hat{y}} = \mathbf{0}, \tag{4.4}$$

with

$$\hat{\mathbf{U}} = \begin{pmatrix} \rho \\ \rho\hat{u} \\ \rho\hat{v} \\ E \end{pmatrix} = \begin{pmatrix} \hat{U}_1 \\ \hat{U}_2 \\ \hat{U}_3 \\ \hat{U}_4 \end{pmatrix}, \qquad \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}.$$

Let $\mathbf{A}(\hat{\mathbf{U}})$ be the Jacobian matrix of $\mathbf{F}(\hat{\mathbf{U}})_{\hat{x}}$,

$$\mathbf{A}(\hat{\mathbf{U}}) = \mathbf{F}'(\hat{\mathbf{U}}) = \begin{pmatrix} \mathbf{a}_1(\hat{\mathbf{U}}) \\ \mathbf{a}_2(\hat{\mathbf{U}}) \\ \mathbf{a}_3(\hat{\mathbf{U}}) \\ \mathbf{a}_4(\hat{\mathbf{U}}) \end{pmatrix}.$$

Since (4.4) is hyperbolic, the matrix $\mathbf{A}(\hat{\mathbf{U}})$ is diagonalized $\mathbf{A}(\hat{\mathbf{U}}) = \mathbf{L}^{-1}\mathbf{\Lambda}\mathbf{L}$ with

$$\mathbf{\Lambda}(\hat{\mathbf{U}}) = diag\{\hat{u} - c, \hat{u}, \hat{u}, \hat{u} + c\}, \quad c = \sqrt{\gamma p/\rho}, \quad \mathbf{L}(\hat{\mathbf{U}}) = \begin{pmatrix} \mathbf{l}_1(\hat{\mathbf{U}}) \\ \mathbf{l}_2(\hat{\mathbf{U}}) \\ \mathbf{l}_3(\hat{\mathbf{U}}) \\ \mathbf{l}_4(\hat{\mathbf{U}}) \end{pmatrix}.$$

The number of boundary conditions required at point P_b depends on the sign of eigenvalues $\{\hat{u} - c, \hat{u}, \hat{u}, \hat{u} + c\}$.

Case 1. $\hat{u} - c \geq 0$: no boundary condition is needed.

Case 2. $\hat{u} - c < 0 \leq \hat{u}$: only one boundary condition is needed.

Case 3. $\hat{u} < 0 \leq \hat{u} + c$: three boundary conditions are needed.

Case 4. $\hat{u} + c < 0$: four boundary conditions are needed.

In the following, we will take case 3 as an example to describe our SILW procedure, assuming that the given boundary conditions are Dirichlet boundary conditions at P_b,

$$\hat{U}_1 = g_1(t), \quad \hat{U}_2 = g_2(t), \quad \hat{U}_3 = g_3(t).$$

For the 2D problem, we will apply the 1D algorithm along $\hat{x}$-direction with the help of the local coordinate transformation. We should point out that, suppose in 1D we take $\alpha \in [\alpha_1, \alpha_2]$ for stability, it should be $\alpha \in [\alpha_1, \alpha_2 \frac{\min(\Delta x, \Delta y)}{h}]$ in 2D. In order to ensure the interval is not null, for 2D cases we choose $k = 4$ and $k_d = 2$, which provides a larger stability region. Thus, we need to construct values on three artificial points, and the following values via ILW procedure

$$\hat{\mathbf{U}}^* = \hat{\mathbf{U}}|_{P_b} + \mathcal{O}(h^5), \quad \hat{\mathbf{U}}_{\hat{x}}^* = \hat{\mathbf{U}}_{\hat{x}}|_{P_b} + \mathcal{O}(h^4), \quad \hat{\mathbf{U}}_{\hat{y}}^* = \hat{\mathbf{U}}_{\hat{y}}|_{P_b} + \mathcal{O}(h^4). \qquad (4.5)$$

In particular, with the local characteristic decomposition, we will use the ILW procedure for the ingoing characteristic variables and extrapolation for the outgoing characteristic variables.

In the local characteristic decomposition, we need to give the approximation of $\hat{\mathbf{U}}$ at P_b at first. Define a quartic polynomial vector $\mathbf{P}(x, y)$ as the approximating polynomial obtained through least squares based on internal point values and derivatives with sufficient grid points, and we can obtain $\hat{\mathbf{P}}(\hat{x}, \hat{y})$ through the rotation transformation (4.3). Then, we can define $\hat{\mathbf{U}}_{ext} = \hat{\mathbf{P}}|_{P_b}$ and perform a local characteristic decomposition $\mathbf{V} = \mathbf{L}(\hat{\mathbf{U}}_{ext})\hat{\mathbf{U}} = (V_1, V_2, V_3, V_4)^T$ for the points near the point P_b. Note that V_4 is the outgoing characteristic variable. Thus, we can get its values and derivatives at P_b by extrapolation (or HWENO extrapolation) directly.

Combining the given boundary conditions and extrapolation on V_4, we can obtain the following system of equations,

$$\begin{cases} \hat{U}_1^* = g_1(t), \\ \hat{U}_2^* = g_2(t), \\ \hat{U}_3^* = g_3(t), \\ \mathbf{l}_4(\hat{\mathbf{U}}_{ext}) \cdot \hat{\mathbf{U}}^* = V_4^*, \end{cases} \tag{4.6}$$

where $V_4^* = \mathbf{l}_4(\hat{\mathbf{U}}_{ext}) \cdot \hat{\mathbf{U}}_{ext}$. Solving the above system gives us the point value $\hat{\mathbf{U}}^*$.

Next we try to find the spatial derivatives $\hat{\mathbf{U}}_{\hat{x}}^*$ by the ILW procedure. We can place the term $\mathbf{G}(\hat{\mathbf{U}})_{\hat{y}}$ in (4.4) as a source term on the right hand side,

$$\hat{\mathbf{U}}_t + \mathbf{A}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{x}} = \mathbf{Resy}, \quad \mathbf{Resy} = -\frac{\partial \mathbf{G}(\hat{\mathbf{U}})}{\partial \hat{y}} = \begin{pmatrix} Resy_1 \\ Resy_2 \\ Resy_3 \\ Resy_4 \end{pmatrix}. \tag{4.7}$$

Plugging the boundary conditions in the first three equations of (4.7) and combining with the extrapolated values $(V_4^*)_{\hat{x}}$ gives us that

$$\begin{cases} \mathbf{a}_1(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{x}}^* = -g_1'(t) + Resy_1, \\ \mathbf{a}_2(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{x}}^* = -g_2'(t) + Resy_2, \\ \mathbf{a}_3(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{x}}^* = -g_3'(t) + Resy_3, \\ \mathbf{l}_4(\hat{\mathbf{U}}_{ext}) \cdot \hat{\mathbf{U}}_{\hat{x}}^* = (V_4)_{\hat{x}}^*. \end{cases} \tag{4.8}$$

Here, all values in $Resy_1, Resy_2, Resy_3$ and the derivatives of the outgoing variable $(V_4)_{\hat{x}}^*$ are obtained by extrapolation polynomial $\hat{\mathbf{P}}$. Hence, we can get $\hat{\mathbf{U}}_{\hat{x}}^*$ by solving the above system.

Similarly, equation (4.4) can also be rewritten as

$$\hat{\mathbf{U}}_t + \mathbf{B}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{y}} = \mathbf{Resx}, \tag{4.9}$$

with

$$\mathbf{B}(\hat{\mathbf{U}}) = \mathbf{G}'(\hat{\mathbf{U}}) = \begin{pmatrix} \mathbf{b}_1(\hat{\mathbf{U}}) \\ \mathbf{b}_2(\hat{\mathbf{U}}) \\ \mathbf{b}_3(\hat{\mathbf{U}}) \\ \mathbf{b}_4(\hat{\mathbf{U}}) \end{pmatrix}, \quad \mathbf{Resx} = -\frac{\partial \mathbf{F}(\hat{\mathbf{U}})}{\partial \hat{x}} = \begin{pmatrix} Resx_1 \\ Resx_2 \\ Resx_3 \\ Resx_4 \end{pmatrix}.$$

Then, $\hat{\mathbf{U}}_{\hat{y}}^*$ can be obtained by solving the linear system

$$\begin{cases} \mathbf{b}_1(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{y}}^* = -g_1'(t) + Resx_1, \\ \mathbf{b}_2(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{y}}^* = -g_2'(t) + Resx_2, \\ \mathbf{b}_3(\hat{\mathbf{U}}^*) \cdot \hat{\mathbf{U}}_{\hat{y}}^* = -g_3'(t) + Resx_3, \\ \mathbf{l}_4(\hat{\mathbf{U}}_{ext}) \cdot \hat{\mathbf{U}}_{\hat{y}}^* = (V_4)_{\hat{y}}^*. \end{cases} \quad (4.10)$$

All values in $Resx_1, Resx_2, Resx_3$ and the derivatives $(V_4)_{\hat{y}}^*$ are obtained by extrapolation as well.

Once we have $\hat{\mathbf{U}}^*$, $\hat{\mathbf{U}}_{\hat{x}}^*$ and $\hat{\mathbf{U}}_{\hat{y}}^*$, we can construct the extrapolation along $\hat{x}$-direction. In particular, we will construct a quartic function $\mathbf{Q}_0(s) = \hat{\mathbf{U}}(P_b + s\mathbf{n}) + \mathcal{O}(h^5)$, and a cubic function $\mathbf{Q}_1(s) = \hat{\mathbf{U}}_{\hat{y}}(P_b + s\mathbf{n}) + \mathcal{O}(h^4)$, which will be employed to evaluate $\{\hat{\mathbf{U}}^*, \hat{\mathbf{U}}_{\hat{x}}^*\}$ and $\hat{\mathbf{U}}_{\hat{y}}^*$ at $P_{i,j}$, respectively.

Algorithm 3. Fifth order SILW method for 2D Euler equations

Step 1. For the ghost point P_{ij} outside the computational domain, we drop a perpendicular to find the foot of the perpendicular P_b on the boundary. We then take the internal point values and derivative values near P_b to perform a least squares fitting to construct $\mathbf{P}(x, y)$. Then, we apply a local coordinate transformation resulting in $\hat{\mathbf{P}}(\hat{x}, \hat{y})$.

Step 2. Construct point value $\hat{\mathbf{U}}^*$ and spatial derivatives $(\hat{\mathbf{U}}^*)_{\hat{x}}, (\hat{\mathbf{U}}^*)_{\hat{y}}$ at P_b via the ILW procedure by solving (4.6), (4.8) and (4.10).

Step 3. Construct informations on three artificial inner points

$$\mathbf{Z}_k^{(0)} = \hat{\mathbf{P}}(P_b - kh\mathbf{n}), \quad \mathbf{Z}_k^{(1)} = \hat{\mathbf{P}}_{\hat{y}}(P_b - kh\mathbf{n}), \quad k = 1, 2, 3.$$

Step 4. Construct a quartic polynomial $\mathbf{Q}_0(s)$ and a cubic polynomial $\mathbf{Q}_1(s)$ satisfying

$$\begin{cases} \mathbf{Q}_0(0) = \hat{\mathbf{U}}^*, \quad \mathbf{Q}_0'(0) = (\hat{\mathbf{U}}^*)_{\hat{x}}, \\ \mathbf{Q}_0(-kh) = \mathbf{Z}_k^{(0)}, \quad k = 1, 2, 3, \\ \mathbf{Q}_1(0) = (\hat{\mathbf{U}}^*)_{\hat{y}}, \\ \mathbf{Q}_1(-kh) = \mathbf{Z}_k^{(1)}, \quad k = 1, 2, 3. \end{cases}$$

Step 5. Take $\bar{s} = |P_{i,j} - P_b|$. Thus, we can obtain the point values and derivative values at $P_{i,j}$ using $\mathbf{Q}_0$ and $\mathbf{Q}_1$,

$$\hat{\mathbf{U}}_{i,j} = \mathbf{Q}_0(\bar{s}), \quad (\hat{\mathbf{U}}_{\hat{x}})_{i,j} = \mathbf{Q}_0'(\bar{s}), \quad (\hat{\mathbf{U}}_{\hat{y}})_{i,j} = \mathbf{Q}_1(\bar{s}).$$

Then, we can obtain $\mathbf{U}_{i,j}$, $(\mathbf{U}_x)_{i,j}$ and $(\mathbf{U}_y)_{i,j}$ by applying a rotation transformation back to (x, y).

5   Numerical tests

In this section, we will validate the effectiveness and stability of our SILW method through several numerical experiments. We will employ the fifth-order HWENO-R scheme for spatial discretization, coupled with the SILW method for boundary treatment. Specifically, for one-dimensional tests, we adopt $(k_d, k) = (2, 3)$, while for two-dimensional tests, we use $(k_d, k) = (2, 4)$. For all cases, we set the parameter $\alpha = 1$. Time discretization is performed using a third-order RK scheme (2.6). The time step is chosen as:

$$\Delta t = 0.6 \frac{\Delta x^k}{a_x}$$

for one-dimensional problems, and

$$\Delta t = \frac{0.6}{a_x/\Delta x^k + a_y/\Delta y^k}$$

for two-dimensional problems. Here, $a_x = \max |\lambda(\mathbf{F}'(\mathbf{U}))|$, $a_y = \max |\lambda(\mathbf{G}'(\mathbf{u}))|$, where λ represents the eigenvalues of the Jacobian matrix. To ensure fifth-order accuracy, $k = 5/3$ for the accuracy tests. Otherwise, we just take $k = 1$.

5.1   One dimensional cases

Example 1. At first, we consider the 1D Burgers' equation

$$\begin{cases} u_t + (\frac{u^2}{2})_x = 0, & x \in (0, 2), \ t > 0, \\ u(x, 0) = 1 + \sin(\pi x), & x \in [0, 2], \\ u(0, t) = g(t), & t > 0. \end{cases}$$

The left boundary $x = 0$ is an inflow boundary, where the boundary condition $g(t)$ is precisely solved using the Newton iteration method for the problem with periodic boundaries, and the time derivative at the boundary is obtained through interpolation. The right boundary $x = 2$ is an outflow boundary, where no boundary condition is needed. We divide the domain with uniform mesh size $\Delta x = 2/N$ and set $C_b = 1 - C_a$.

At $T = 0.5/\pi$, the exact solution is smooth, and the corresponding L^1 errors and L^∞ errors of u are listed in Tables 5.2 and 5.3, respectively. We can observe that our scheme is always stable and can achieve the designed fifth order for different C_a. A discontinuity appears in the exact solution at $T = 1.5/\pi$, and the discontinuity has propagated through the left boundary by $T = 5/\pi$. The numerical solutions with $N = 80$ and exact solutions at these times are compared in Figure 5.4, showing that the proposed schemes can capture the discontinuity well without noticeable oscillations.

Table 5.2. Example 1: L^1 errors and orders of accuracy of u at $T = 0.5/\pi$.

N	$C_a = 0.01$		$C_a = 0.50$		$C_a = 0.99$	
	L^1 error	order	L^1 error	order	L^1 error	order
10	4.274E-02	-	2.558E-02	-	4.099E-02	-
20	3.543E-03	3.59	6.995E-03	1.87	2.976E-03	3.78
40	5.591E-05	5.99	4.561E-04	3.94	9.726E-05	4.94
80	1.838E-06	4.93	1.515E-05	4.91	1.949E-06	5.64
160	2.393E-08	6.26	3.791E-08	8.64	2.091E-08	6.54
320	5.928E-10	5.34	5.119E-10	6.21	5.054E-10	5.37
640	1.739E-11	5.09	1.462E-11	5.13	1.490E-11	5.08

Table 5.3. Example 1: L^∞ errors and orders of accuracy of u at $T = 0.5/\pi$.

N	$C_a = 0.01$		$C_a = 0.50$		$C_a = 0.99$	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10	8.752E-02	-	8.223E-02	-	8.421E-02	-
20	1.771E-02	2.30	1.859E-02	2.15	1.470E-02	2.52
40	2.542E-04	6.12	3.372E-03	2.46	7.600E-04	4.27
80	2.369E-05	3.42	1.687E-04	4.32	3.704E-05	4.36
160	2.215E-07	6.74	5.792E-07	8.19	2.216E-07	7.38
320	6.338E-09	5.13	6.340E-09	6.51	6.339E-09	5.13
640	1.890E-10	5.07	1.889E-10	5.07	1.890E-10	5.07

Example 2. Here, we test the one-dimensional linear system

$$\begin{cases} u_t + w_x = 0, & x \in (0, 2\pi), \ t > 0, \\ w_t + u_x = 0, & x \in (0, 2\pi), \ t > 0, \\ u(x, 0) = \sin(x), & w(x, 0) = -\sin(x), \quad x \in [0, 2\pi], \\ u(0, t) = \sin(t), & u(2\pi, t) = \sin(t), \quad t > 0. \end{cases}$$

We take $C_b = 1 - C_a$ and $\Delta x = 2\pi/N$. In this example, only one boundary condition is required at both $x = 0$ and $x = 2\pi$. Here, Dirichlet boundary conditions for u are imposed. The exact solution is $u(x, t) = -w(x, t) = \sin(x + t)$. We provide the L^1 error in Table 5.4 and the L^∞ error in Table 5.5 for u at $T = 1$ with respect to different values
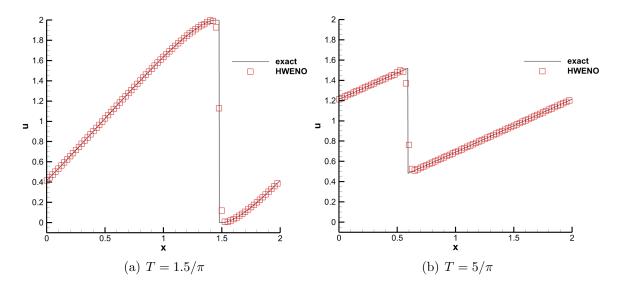
(a) $T = 1.5/\pi$ (b) $T = 5/\pi$

Figure 5.4. Example 1: solutions with $N = 80$, $C_a = 0.01$.

of C_a, demonstrating that the scheme is stable and can reach the designed high order for linear system.

Table 5.4. Example 2: L^1 errors and orders of accuracy of u at $T = 1$.

N	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^1 error	order	L^1 error	order	L^1 error	order
10	4.609E-02	-	2.870E-02	-	4.606E-002	-
20	1.503E-03	4.94	1.417E-03	4.34	1.519E-03	4.92
40	5.957E-05	4.66	4.604E-05	4.94	2.040E-05	6.22
80	1.945E-06	4.94	1.197E-06	5.27	1.272E-06	4.00
160	4.465E-08	5.45	9.314E-09	7.01	2.298E-08	5.79
320	1.526E-09	4.87	2.903E-10	5.00	7.882E-10	4.87
640	5.115E-11	4.90	9.772E-12	4.89	2.701E-11	4.87

Table 5.5. Example 2: L^∞ errors and orders of accuracy of u at $T = 1$.

N	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10	9.573E-02	-	7.534E-02	-	1.571E-01	-
20	4.987E-03	4.26	3.100E-03	4.60	1.364E-02	3.53
40	2.775E-04	4.17	1.955E-04	3.99	1.416E-04	6.59
80	1.894E-05	3.87	9.778E-06	4.32	8.972E-06	3.98
160	3.060E-07	5.95	8.292E-08	6.88	1.482E-07	5.92
320	1.027E-08	4.90	2.944E-09	4.82	5.553E-09	4.74
640	3.384E-10	4.92	1.009E-10	4.87	1.969E-10	4.82

Example 3. Next, we consider a 1D Euler equation on $[0, 2\pi]$ with initial data

$$\begin{cases} \rho(x,0) = 1 + 0.2\sin(x), \\ p(x,0) = 2, \\ u(x,0) = 1. \end{cases}$$

The grid partitioning is the same as in Example 2. Boundary conditions on ρ and u at $x = 0$ and ρ at $x = 2\pi$ are given such that the system has an exact solution $\rho(x,t) = 1 + 0.2\sin(x - t), p(x,t) = 2, u(x,t) = 1$. At $T = 1$, the errors and orders of accuracy of the density ρ are provided in Tables 5.6 and 5.7, indicating that our methods can achieve the designed fifth order accuracy.

Table 5.6. Example 3: L^1 errors and orders of accuracy of ρ at $T = 1$.

N	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^1 error	order	L^1 error	order	L^1 error	order
10	3.488E-03	-	1.585E-03	-	4.784E-03	-
20	1.481E-04	4.56	2.225E-04	2.83	9.988E-05	5.58
40	2.998E-06	5.63	1.042E-06	7.74	2.725E-06	5.20
80	7.021E-08	5.42	1.662E-08	5.97	3.382E-08	6.33
160	2.143E-09	5.03	5.104E-10	5.03	1.268E-09	4.74
320	7.642E-11	4.81	1.735E-11	4.88	4.635E-11	4.77
640	2.795E-12	4.77	7.551E-13	4.52	1.842E-12	4.65

Table 5.7. Example 3: L^∞ errors and orders of accuracy of ρ at $T = 1$.

N	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10	1.417E-02	-	3.494E-03	-	1.972E-02	-
20	1.853E-03	2.93	8.817E-04	1.99	3.495E-04	5.82
40	6.883E-05	4.75	4.081E-06	7.76	2.177E-05	4.00
80	1.358E-06	5.66	2.262E-07	4.32	6.742E-07	5.01
160	3.630E-08	5.22	7.956E-09	4.83	2.606E-08	4.69
320	1.419E-09	4.68	2.980E-10	4.74	9.530E-10	4.77
640	5.128E-11	4.79	1.044E-11	4.83	3.251E-11	4.87

Example 4. Then we consider the interaction of two blast waves. In this problem, multiple reflections occur from shock waves and rarefaction waves on the wall. The initial conditions are

$$(\rho, u, p) = \begin{cases} (1, 0, 1000), & x \in [0, 0.1), \\ (1, 0, 0.01), & x \in [0.1, 0.9), \\ (1, 0, 100), & x \in [0.9, 1.0]. \end{cases}$$

We apply reflective boundary conditions at both boundaries and compute the numerical solution at $T = 0.038$. The reference solution is computed using the fifth order WENO scheme on body-fitted uniform mesh with $\Delta x = 1/16000$. Numerical results with different mesh size are provided in Figure 5.5. We can see that the proposed scheme can distinguish the structure of the solution well.
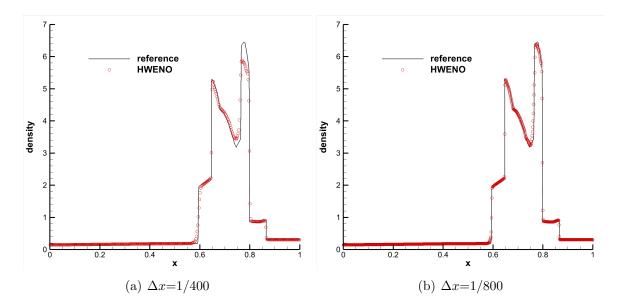

(a) $\Delta x = 1/400$ (b) $\Delta x = 1/800$

Figure 5.5. Example 4: the density profiles of the blast wave problem.

5.2 Two dimensional cases

Example 5. At first, we look at the 2D Burgers' equation on a rectangular domain

$$
\begin{cases}
u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, & (x,y) \in [0,4] \times [0,4], \quad t > 0, \\
u(x,y,0) = 0.75 + 0.5\sin(\pi\left(\frac{x+y}{2}\right)), & (x,y) \in [0,4] \times [0,4].
\end{cases}
$$

Here we define $x_i = (C_a - 1 + i)\Delta x$, $y_i = (C_a - 1 + i)\Delta y$, $1 \le i \le N$, where $C_a \in [0,1)$ and $\Delta x = \Delta y = 4/N$. The boundaries $x = 0$ and $y = 0$ are inflow boundaries, where the boundary conditions obtained from the exact solution with periodic boundaries are imposed here. On the other hand, $x = 4$ and $y = 4$ are outflow boundaries, which require no boundary conditions. The solution remains continuous at $T = 1/\pi$, and the corresponding errors are listed in Tables 5.8 and 5.9. Both the 1D cross-section along the diagonal and the two-dimensional plane at $T = 6/\pi$ are presented in the figure 5.6, when discontinuities exist. We can see that our 2D method can achieve the optimal fifth order for smooth solutions and avoid numerical oscillation near discontinuities.

22

Table 5.8. Example 5: L^1 errors and orders of accuracy of u at $T = 1/\pi$.

$N_x \times N_y$	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^1 error	order	L^1 error	order	L^1 error	order
10×10	1.113E-02	-	1.735E-02	-	6.862E-03	-
20×20	8.105E-04	3.78	6.016E-04	4.85	5.454E-04	3.65
40×40	3.367E-05	4.59	9.622E-06	5.97	2.836E-05	4.27
80×80	6.919E-07	5.60	6.507E-07	3.89	4.063E-07	6.13
160×160	1.114E-08	5.96	1.067E-08	5.93	1.017E-08	5.32
320×320	3.158E-10	5.14	3.101E-10	5.10	2.919E-10	5.12
640×640	9.349E-12	5.08	9.179E-12	5.08	8.858E-12	5.04

Table 5.9. Example 5: L^∞ errors and orders of accuracy of u at $T = 1/\pi$

$N_x \times N_y$	$C_a = 0.01$		$C_a = 0.5$		$C_a = 0.99$	
	L^∞ error	order	L^∞ error	order	L^∞ error	order
10×10	4.094E-02	-	3.722E-02	-	1.983E-02	-
20×20	2.896E-03	3.82	2.449E-03	3.93	2.852E-03	2.80
40×40	1.548E-04	4.23	8.994E-05	4.77	1.528E-04	4.22
80×80	1.003E-05	3.95	7.778E-06	3.53	4.573E-06	5.06
160×160	1.311E-07	6.26	1.281E-07	5.92	1.307E-07	5.13
320×320	3.857E-09	5.09	3.876E-09	5.05	3.852E-09	5.08
640×640	1.161E-10	5.05	1.168E-10	5.05	1.161E-10	5.05

(a) Cut of the solution along $x = y$.
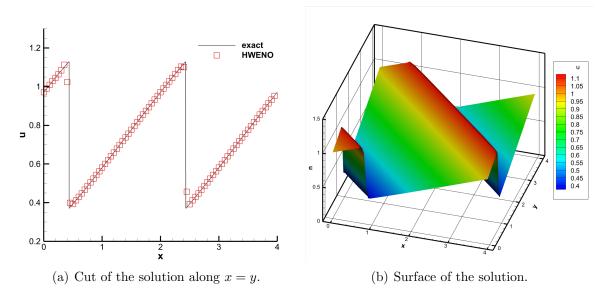
(b) Surface of the solution.

Figure 5.6. Example 5: Solutions with $N_x \times N_y = 80 \times 80$, $C_a = 0.01$, $T = 6/\pi$.

Example 6. Next, we consider the 2D Burgers' equation on a disk,

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

with

$$\Omega = \{(x, y) \mid x^2 + y^2 \le 4\}.$$

Here, we define $x_i = i\Delta x$ and $y_i = i\Delta y$. We impose the problem with Dirichlet boundary conditions on the inflow boundary such that the exact solution is the same as that of Example 5. The errors of the numerical solution at $t = 1/\pi$ are provided in Table 5.10, and the plots along the diagonal as well as the 2D plane are shown in Figure 5.7. It is observed that our scheme is effective for curved boundary.

Table 5.10. Example 6: errors and orders of accuracy of u at $T = 1/\pi$.

$\Delta x = \Delta y$	L^1 error	order	L^∞ error	order
1/5	1.173E-03	-	4.202E-03	-
1/10	5.161E-05	4.51	2.727E-04	3.95
1/20	1.347E-06	5.26	2.129E-05	3.68
1/40	1.571E-08	6.42	1.309E-07	7.35
1/80	4.514E-10	5.12	8.986E-09	3.86
1/160	1.284E-11	5.14	2.444E-10	5.20

(a) Cut of the solution along $x = y$.　　　(b) Surface of the solution.
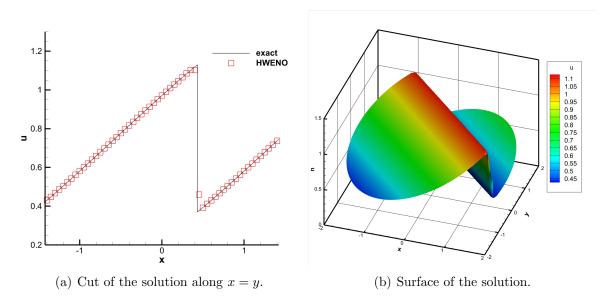
Figure 5.7. Example 6: solutions with $\Delta x = \Delta y = 1/20$ at $T = 6/\pi$.

Example 7. We use an example with a curved wall to test our schemes. The initialization of the problem is a Mach 3 flow moving from the left to a unit radius cylinder located at the origin of the $x - y$ plane. Due to the symmetry of the problem, we only need to compute the upper half plane, i.e., $[-3, 0] \times [0, 6]$. On the surface of the cylinder, we use the ILW method to deal with the solid-wall boundary conditions. At the left boundary of the computed region $x = -3$, the inflow boundary conditions are given.

And for the right boundary $x = 0$ and the upper boundary $y = 6$, outflow boundary conditions are imposed. Pressure and Mach number are given in Figure 5.8, which are comparable with those in the previous work.
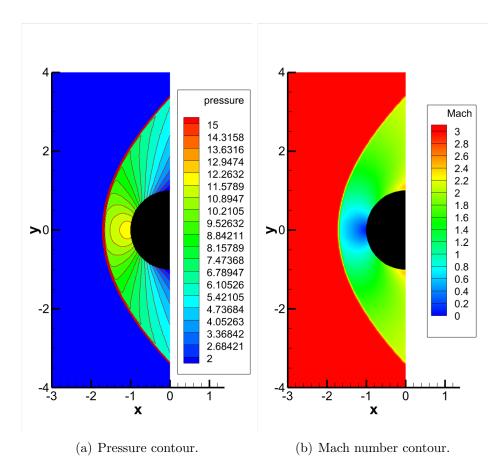

(a) Pressure contour.   (b) Mach number contour.

Figure 5.8. Example 7: pressure and Mach number of flow past a cylinder with $\Delta x = \Delta y = 1/40$.

Example 8. Finally, we test the double Mach reflection problem. This example describes the interaction between a shock wave with a Mach number of 10 moving in the horizontal direction and a solid wall at an angle of $30°$. The domain is shown in Figure 5.9. At the top boundary of the computational domain, we set appropriate boundary conditions based on the given Mach number, while supersonic inflow and outflow boundary conditions are applied on the left and right sides, respectively. For the hypotenuse solid wall, the designed ILW method is employed. Density contours with different mesh sizes are presented in Figure 5.10, and the zoomed-in regions near the double Mach stem are also given. It is observed that the results are comparable with the previous results, indicating the effectiveness of our ILW method.
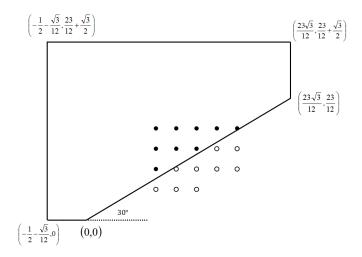
25

Figure 5.9. Example 8: double Mach reflection computational region

6 Conclusion

In this paper, we introduce a novel ILW boundary treatment method for fifth-order finite difference HWENO schemes [5] to solve hyperbolic conservation laws. Distinct from existing ILW approaches [16, 22, 23], our method incorporates additional parameters k and employs a least squares method. By adjusting the parameter k, our method could achieves stability while reducing the required terms with ILW procedure. Therefore our scheme enhances computational efficiency, particularly for multi-dimensional systems. Additionally, we provide a stability analysis for the fully discrete schemes, specifying appropriate parameter values to ensure the boundary treatment does not effect the original CFL condition constraint of the inner scheme, thus it can avoid the cut-cell problem effectively. Through numerical experiments, we demonstrate that our algorithm is stable, robust, and achieves the designed fifth-order accuracy, performing exceptionally well in complex problems such as cylindrical flow and double Mach reflection problems.

A Fifth order HWENO algorithm

Here, we give the formulation of the fifth order HWENO scheme to construct $\hat{f}_{i+1/2}^{+}$ and $\hat{h}_{i+1/2}^{+}$ based on $\{(f_j^{+}, h_j^{+}), j = i-1, i, i+1\}$ [5].

1. Construct a quartic polynomial $p_0(x)$ and two quadratic polynomials $p_1(x), p_2(x)$, satisfying

$$\begin{cases} \dfrac{1}{\Delta x} \displaystyle\int_{I_{i+l}} p_0(x)dx = f_{i+l}^{+}, & l = -1, 0, 1, \\ \dfrac{1}{\Delta x} \displaystyle\int_{I_{i+l}} p_0'(x)dx = h_{i+l}^{+}, & l = -1, 1. \end{cases}$$

26

(a) $\Delta x = \Delta y = 1/320$

(b) $\Delta x = \Delta y = 1/320$, zoomed-in figure.

(c) $\Delta x = \Delta y = 1/640$

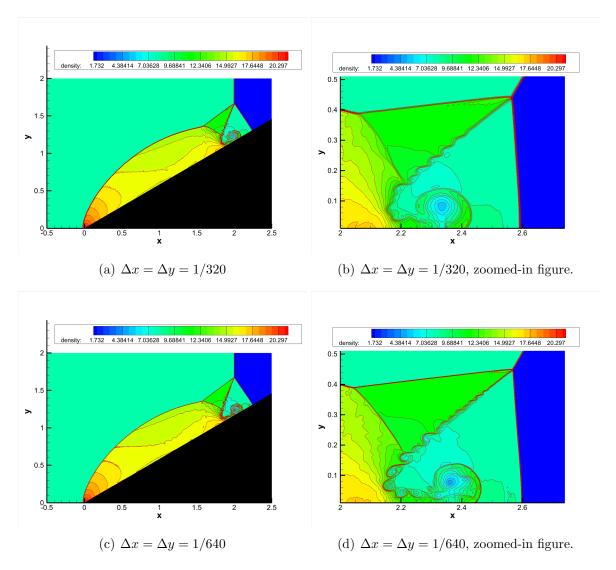(d) $\Delta x = \Delta y = 1/640$, zoomed-in figure.

Figure 5.10. Example 8: density contour of double Mach reflection, 30 contours from 1.731 to 20.92.

$$\frac{1}{\Delta x} \int_{I_{i+l}} p_1(x)dx = f_{i+l}^+, \quad l = -1, 0,$$

$$\frac{1}{\Delta x} \int_{I_{i+l}} p_2(x)dx = f_{i+l}^+, \quad l = 0, 1.$$

Compute the values and derivatives at half point $x_{i+\frac{1}{2}}$:

$$p_0(x_{i+\frac{1}{2}}) = -\frac{23}{120}f_{i-1}^+ + \frac{19}{30}f_i^+ + \frac{67}{120}f_{i+1}^+ - \frac{3\Delta x}{40}h_{i-1}^+ - \frac{7\Delta x}{40}h_{i+1}^+,$$

$$p_1(x_{i+\frac{1}{2}}) = -\frac{1}{2}f_{i-1}^+ + \frac{3}{2}f_i^+,$$

$$p_2(x_{i+\frac{1}{2}}) = \frac{1}{2}f_i^+ + \frac{1}{2}f_{i+1}^+,$$

$$p_0'(x_{i+\frac{1}{2}}) = \frac{1}{\Delta x}\left(\frac{3}{8}f_{i-1}^+ - 2f_i^+ + \frac{13}{8}f_{i+1}^+\right) + \frac{1}{8}h_{i-1}^+ - \frac{3}{8}h_{i+1}^+.$$

Define the linear weights γ_l, $l = 0, 1, 2$, satisfying

$$\gamma_l \geq 0, \quad \text{and} \quad \gamma_0 + \gamma_1 + \gamma_2 = 1.$$

Here, we take $\gamma_0 = 0.99, \gamma_1 = \gamma_2 = 0.005$.

2. Compute the smoothness indicator β_l at interval I_i to measure the smoothness of the polynomial $p_l(x)$, $l = 0, 1, 2$,

$$\beta_l = \sum_{\alpha=1}^{k} \int_{I_i} \Delta x^{2\alpha-1} \left(\frac{d^\alpha p_l(x)}{dx^\alpha}\right)^2 dx, \quad l = 0, 1, 2.$$

3. Form the nonlinear weight ω_l based on the linear weight γ_l and the smooth indicator β_l:

$$\omega_l = \frac{\bar{\omega}_l}{\sum_{l=0}^{2}\bar{\omega}_l}, \quad \bar{\omega}_l = \gamma_l\left(1 + \frac{\tau}{\beta_l + \epsilon}\right), \quad l = 0, 1, 2$$

with

$$\tau = \frac{1}{4}(|\beta_0 - \beta_1| + |\beta_0 - \beta_2|)^2.$$

And $\epsilon = 10^{-10} > 0$ is a tiny number to avoid the denominator to become 0.

4. Finally, we reconstruct $\hat{f}_{i+\frac{1}{2}}^+$ with nonlinear weights and $\hat{h}_{i+\frac{1}{2}}^+$ with linear weights:

$$\hat{f}_{i+\frac{1}{2}}^+ = \omega_0\left(\frac{1}{\gamma_0}p_0(x_{i+\frac{1}{2}}) - \frac{\gamma_1}{\gamma_0}p_1(x_{i+\frac{1}{2}}) - \frac{\gamma_2}{\gamma_0}p_2(x_{i+\frac{1}{2}})\right) + \omega_1 p_1(x_{i+\frac{1}{2}}) + \omega_2 p_2(x_{i+\frac{1}{2}}),$$

$$\hat{h}_{i+\frac{1}{2}}^+ = p_0'(x_{i+\frac{1}{2}})$$

B Modification of the derivative

Here, we present the modification on the derivative v_i [5] in the following.

1. Firstly, construct a quartic polynomial $q_0(x)$ and two first-order polynomial $q_1(x), q_2(x)$:

$$\begin{cases} q_0(x_{i+l}) = u_{i+l}, & l = -1, 0, 1, \\ q_0'(x_{i+l}) = v_{i+l}, & l = -1, 1. \end{cases}$$

$$q_1(x_{i+l}) = u_{i+l}, \quad l = -1, 0,$$

$$q_2(x_{i+l}) = u_{i+l}, \quad l = 0, 1.$$

Then we have

$$q_0'(x_i) = \frac{3}{4\Delta x}(u_{i+1} - u_{i-1}) - \frac{1}{4}(v_{i-1} + v_{i+1}),$$

$$q_1'(x_i) = \frac{1}{\Delta x}(u_i - u_{i-1}),$$

$$q_2'(x_i) = \frac{1}{\Delta x}(u_{i+1} - u_i).$$

2. Compute the smooth indicator β_l of the polynomial $q_0(x), q_1(x), q_2(x)$.

$$\beta_l = \sum_{\alpha=1}^{k} \int_{I_i} \Delta x^{2\alpha-1} \left(\frac{d^\alpha q_l(x)}{dx^\alpha} \right)^2 dx, \quad l = 0, 1, 2.$$

3. Define the nonliear weights

$$\lambda_l = \frac{\bar{\lambda}_l}{\sum_{l=0}^{2} \bar{\lambda}_l}, \quad \bar{\lambda}_l = d_l \left(1 + \frac{\tau}{\beta_l + \epsilon} \right), \quad l = 0, 1, 2$$

with linear weights $d_0 = 0.9, d_1 = d_2 = 0.05$ and

$$\tau = \frac{1}{4}(|\beta_0 - \beta_1| + |\beta_0 - \beta_2|)^2.$$

4. Finally, obtain the modification of the first derivative at x_i as

$$\tilde{v}_i = \lambda_0 \left(\frac{1}{d_0}q_0'(x_i) - \frac{d_1}{d_0}q_1'(x_i) - \frac{d_2}{d_0}q_2'(x_i) \right) + \lambda_1 q_1'(x_i) + \lambda_2 q_2'(x_i).$$

References

[1] M. H. Carpenter, D. Gottlieb, S. Abarbanel, and W.-S. Don. The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM Journal on Scientific Computing*, 16(6):1241–1252, 1995.

[2] Z. Cheng, S. Liu, Y. Jiang, J. Lu, M. Zhang, and S. Zhang. A high order boundary scheme to simulate complex moving rigid body under impingement of shock wave. *Applied Mathematics and Mechanics*, 42(6):841–854, 2021.

[3] Z. Du and J. Li. A Hermite WENO reconstruction for fourth order temporal accurate schemes based on the GRP solver for hyperbolic conservation laws. *Journal of Computational Physics*, 355:385–396, 2018.

[4] C. Fan and K. Wu. High-order oscillation-eliminating Hermite WENO method for hyperbolic conservation laws. *Journal of Computational Physics*, 519:113435, 2024.

[5] C. Fan, Z. Zhao, T. Xiong, and J. Qiu. A robust fifth order finite difference Hermite WENO scheme for compressible Euler equations. *Computer Methods in Applied Mechanics and Engineering*, 412:116077, 2023.

[6] F. Filbet and C. Yang. An inverse Lax–Wendroff method for boundary conditions applied to Boltzmann type models. *Journal of Computational Physics*, 245:43–61, 2013.

[7] J. Li, C.-W. Shu, and J. Qiu. Multi-resolution HWENO schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 446:110653, 2021.

[8] L. Li, J. Zhu, C.-W. Shu, and Y.-T. Zhang. A fixed-point fast sweeping WENO method with inverse Lax-Wendroff boundary treatment for steady state of hyperbolic conservation laws. *Communications on Applied Mathematics and Computation*, 5(1):403–427, 2023.

[9] T. Li, J. Lu, and C.-W. Shu. Stability analysis of inverse Lax–Wendroff boundary treatment of high order compact difference schemes for parabolic equations. *Journal of Computational and Applied Mathematics*, 400:113711, 2022.

[10] 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. *Journal of Computational and Applied Mathematics*, 299:140–158, 2016.

[11] T. Li, C.-W. Shu, and M. Zhang. Stability analysis of the inverse Lax–Wendroff boundary treatment for high order central difference schemes for diffusion equations. *Journal of Scientific Computing*, 70:576–607, 2017.

[12] H. Liu and J. Qiu. Finite difference Hermite WENO schemes for hyperbolic conservation laws. *Journal of Scientific Computing*, 63(2):548–572, 2015.

[13] H. Liu and J. Qiu. Finite difference Hermite WENO schemes for conservation laws, II: an alternative approach. *Journal of Scientific Computing*, 66(2):598–624, 2016.

[14] S. Liu, Z. Cheng, Y. Jiang, J. Lu, M. Zhang, and S. Zhang. Numerical simulation of a complex moving rigid body under the impingement of a shock wave in 3d. *Advances in Aerodynamics*, 4(1):8, 2022.

[15] S. Liu, Y. Jiang, C.-W. Shu, M. Zhang, and S. Zhang. A high order moving boundary treatment for convection-diffusion equations. *Journal of Computational Physics*, 473:111752, 2023.

[16] S. Liu, T. Li, Z. Cheng, Y. Jiang, C.-W. Shu, and M. Zhang. A new type of simplified inverse Lax-Wendroff boundary treatment I: Hyperbolic conservation laws. *Journal of Computational Physics*, 514:113259, 2024.

[17] Z. Liu, Y. Jiang, C.-W. Shu, and M. Zhang. High-order inverse Lax-Wendroff procedure for compressible fluid-structure interaction problems. *Journal of Computational Physics*, 531:113942, 2025.

[18] J. Lu, J. Fang, S. Tan, C.-W. Shu, and M. Zhang. Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations. *Journal of Computational Physics*, 317:276–300, 2016.

[19] C. S. Peskin. Flow patterns around heart valves: a numerical method. *Journal of computational physics*, 10(2):252–271, 1972.

[20] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193(1):115–135, 2004.

[21] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: Two dimensional case. *Computers & Fluids*, 34(6):642–663, 2005.

[22] S. Tan and C.-W. Shu. Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *Journal of Computational Physics*, 229(21):8144–8166, 2010.

[23] S. Tan, C. Wang, C.-W. Shu, and J. Ning. Efficient implementation of high order inverse Lax–Wendroff boundary treatment for conservation laws. *Journal of Computational Physics*, 231(6):2510–2527, 2012.

[24] F. Vilar and C.-W. Shu. Development and stability analysis of the inverse Lax-Wendroff boundary treatment for central compact schemes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(1):39–67, 2015.

[25] X. Xie, Z. Tao, C. Jiao, and M. Zhang. An efficient fifth-order interpolation-based Hermite WENO scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 523:113673, 2025.

[26] L. Yang, S. Li, Y. Jiang, C.-W. Shu, and M. Zhang. Inverse Lax-Wendroff boundary treatment of discontinuous Galerkin method for 1D conservation laws. *Communications on Applied Mathematics and Computation*, 7:796–826, 2025.

[27] Y. H. Zahran and A. H. Abdalla. Seventh order Hermite WENO scheme for hyperbolic conservation laws. *Computers & Fluids*, 131:66–80, 2016.

[28] M. Zhang and Z. Zhao. A fifth-order finite difference HWENO scheme combined with limiter for hyperbolic conservation laws. *Journal of Computational Physics*, 472:111676, 2023.

[29] Z. Zhao, Y.-T. Zhang, and J. Qiu. A modified fifth order finite difference Hermite WENO scheme for hyperbolic conservation laws. *Journal of Scientific Computing*, 85(2):29, 2020.

[30] G. Zhu, Y. Jiang, and M. Zhang. Inverse Lax-Wendroff boundary treatment for solving conservation laws with finite volume methods. *Communications on Applied Mathematics and Computation*, 2024.