Inverse Lax-Wendroff procedure for numerical boundary conditions of convection-diffusion equations

Jianfang Lu[1], Jinwei Fang[2], Sirui Tan[3], Chi-Wang Shu[4] and Mengping Zhang[5]

Abstract

We consider numerical boundary conditions for high order finite difference schemes for solving convection-diffusion equations on arbitrary geometry. The two main difficulties for numerical boundary conditions in such situations are: (1) the wide stencil of the high order finite difference operator requires special treatment for a few ghost points near the boundary; (2) the physical boundary may not coincide with grid points in a Cartesian mesh and may intersect with the mesh in an arbitrary fashion. For purely convection equations, the so-called inverse Lax-Wendroff procedure [28], in which we convert the normal derivatives into the time derivatives and tangential derivatives along the physical boundary by using the equations, have been quite successful. In this paper, we extend this methodology to convection-diffusion equations. It turns out that this extension is non-trivial, because totally different boundary treatments are needed for the diffusion-dominated and the convection-dominated regimes. We design a careful combination of the boundary treatments for the two regimes and obtain a stable and accurate boundary condition for general convection-diffusion equations. We provide extensive numerical tests for one- and two-dimensional problems involving both scalar equations and systems, including the compressible Navier-Stokes equations, to demonstrate the good performance of our numerical boundary conditions.

Key Words: convection-diffusion equation, high order finite difference methods, numerical boundary condition, inverse Lax-Wendroff method, compressible Navier-Stokes equations

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

[2]Department of Mathematics, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China. E-mail: gerrard.fang@gmail.com

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

[4]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu. Research supported by AFOSR grant F49550-12-1-0399 and NSF grant DMS-1418750.

[5]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: mpzhang@ustc.edu.cn. Research supported by NSFC grant 11471305.

1 Introduction

The inverse Lax-Wendroff (ILW) method was introduced by Tan and Shu [28] as a technique of numerical boundary condition for solving hyperbolic equations on arbitrary geometry using high order finite difference methods. A stable and high order accurate numerical boundary treatment requires significant efforts for two reasons. First, the wide stencils of high order finite difference operators need special treatments for a few ghost points near the boundary. Second, the physical domain boundary in arbitrary geometry may not coincide with grid points and may intersect with the Cartesian mesh in an arbitrary fashion. There are several approaches to handle such boundary conditions. One commonly used method is to generate a boundary fitted mesh which allows us to impose boundary conditions directly in the algorithm. As a result, the governing equations are generally transformed into a new differential form in a curvilinear coordinate system (see e.g. [13]). If the domain is simple enough, a smooth mapping can be used for the transformation of the entire domain. But if the domain is complex, composite overlapping meshes are generated to fit the physical boundaries, while these meshes are connected via interpolation (see e.g. [4, 10, 9, 25]). The disadvantage of this method is the complexity of generating the body-conforming grids. Especially when the boundary varies with time, the generation of a new mesh at each time level is required, which increases the computational cost tremendously. Another approach is to use Cartesian grids which do not conform to the physical boundary. The embedded boundary method is developed to solve the wave equation with Dirichlet or Neumann boundary conditions by using finite difference methods on Cartesian grids in [15, 16, 14, 19]. In [24] the authors applied this method to hyperbolic conservation laws and obtained a second order accurate scheme. Baeza et al. [2] extended the approach from second order to fifth order using Lagrange extrapolation with a filter for the detection of discontinuities.

A third approach is the so called immersed boundary method, which makes a modification of the original partial differential equations by introducing a forcing function

at the physical boundary, and uses it to reproduce the effects of boundary conditions, see, e.g. [22, 18]. The inverse Lax-Wendroff (ILW) method is similar to the immersed boundary method, but without introducing any forcing function to alter the original equations. In [28, 29, 30, 31], the authors developed this high order accurate boundary treatment for hyperbolic conservation laws, based on the ILW procedure for the inflow boundaries and high order extrapolation for the outflow boundaries. This boundary treatment allows us to compute hyperbolic conservation laws defined on an arbitrarily shaped domain covered by a Cartesian mesh with high order accuracy. The main idea of the ILW procedure is repeatedly using the partial differential equation to convert the normal derivatives into the time derivatives and tangential derivatives at the physical boundary. With the given inflow boundary condition and these normal derivatives, we can use Taylor expansions to assign values to ghost points near the physical boundary. A simplified ILW procedure which uses the ILW process only for the first few normal derivatives and then the less expensive high order extrapolation for the remaining ones is developed in [31]. Stability analysis for both the ILW and the simplified ILW procedures is given in [32, 17], proving that the method remains stable under the same CFL condition as that for problems without boundaries, regardless of the location of the first grid point relative to the location of the physical boundary. This indicates that the "cut cell" problem [1], which refers to the instability or the requirement of the extremely small CFL condition when the first grid point does not coincide with but is very close to the physical boundary, is effectively removed by the ILW or the simplified ILW procedure. For earlier related work, see [6, 7, 11].

So far, the ILW method has been mostly used on hyperbolic equations including conservation laws [28, 29, 31, 17] and Boltzmann type models [5], which involve only first order spatial derivatives in the equations. In this paper, we attempt to extend this methodology to convection-diffusion equations. It turns out that this extension is non-trivial, as totally different boundary treatments are needed for the diffusion-dominated and the

convection-dominated regimes. A careful combination of these two boundary treatments is designed in this paper to obtain a stable and accurate boundary condition for high order finite difference schemes when applied to convection-diffusion equations, regardless of the regimes. We provide extensive numerical tests for one and two-dimensional problems involving both scalar equations and systems, including the compressible Navier-Stokes equations, to demonstrate the good performance of our numerical boundary conditions.

We comment on the additional issues encountered in solving the compressible Navier-Stokes equations. Unlike the model convection-diffusion equations that we consider first, the compressible Navier-Stokes equations form a nonlinear incompletely parabolic system. There are many papers discussing boundary conditions of incompletely parabolic systems based on the energy method, for example [8, 21, 20]. Our approach for the diffusion-dominated regime of the compressible Navier-Stokes equations is based on the theory in [26, 8], which reduces the incompletely parabolic system into a parabolic system and a strictly hyperbolic system.

This paper is organized as follows. In Section 2, we first illustrate our idea by developing our inverse Lax-Wendroff method for one-dimensional scalar convection-diffusion equations. We then generalize it to one-dimensional systems and two-dimensional problems. For the convection-dominated cases with a shock-like sharp transition region near the outflow boundary, we adopt a weighted essentially non-oscillatory (WENO) type extrapolation, see e.g. [28, 31, 12], to maintain nonlinear stability. In Section 3, we show numerical results to demonstrate the effectiveness and robustness of our approach. Concluding remarks are given in section 4.

2 Scheme formulation

We consider a convection-diffusion equation, abstractly written as

$$\boldsymbol{U}_t = L(\boldsymbol{U}), \tag{2.1}$$

where L is a spatial operator involving first and second order spatial derivatives. After the spatial operator is discretized by a high order finite difference operator, the semi-discrete scheme is written abstractly as the following ordinary differential equation (ODE) system

$$\boldsymbol{U}_t = L_h(\boldsymbol{U}). \tag{2.2}$$

Our main interest is in the convection-dominated regime. Thus we use the following third order total variation diminishing (TVD) Runge-Kutta method [23] to discretize (2.2)

$$\begin{aligned}
\boldsymbol{U}^{(1)} &= \boldsymbol{U}^n + \Delta t L_h(\boldsymbol{U}^n), \\
\boldsymbol{U}^{(2)} &= \frac{3}{4}\boldsymbol{U}^n + \frac{1}{4}\boldsymbol{U}^{(1)} + \frac{1}{4}\Delta t L_h(\boldsymbol{U}^{(1)}), \\
\boldsymbol{U}^{n+1} &= \frac{1}{3}\boldsymbol{U}^n + \frac{2}{3}\boldsymbol{U}^{(2)} + \frac{2}{3}\Delta t L_h(\boldsymbol{U}^{(2)}).
\end{aligned}$$

In [3], the authors pointed out that the boundary condition should be imposed as follows in order to maintain the third order accuracy

$$\begin{aligned}
U^n &= g(t_n), \\
U^{(1)} &= g(t_n) + \Delta t g'(t_n), \\
U^{(2)} &= g(t_n) + \frac{\Delta t}{2} g'(t_n) + \frac{\Delta t^2}{4} g''(t_n),
\end{aligned}$$

where $g(t)$ is the boundary condition for the governing equations. We follow this approach in our numerical boundary treatments. For simplicity, in the following we simply use $g(t)$ to denote our boundary condition, without distinguishing its different meaning at different Runge-Kutta stages.

2.1 Inverse Lax-Wendroff method for hyperbolic equations

Let us briefly review the ILW method for hyperbolic equations [28], to explain the basic idea and to set notations. For simplicity, we consider the following one-dimensional linear equation

$$\begin{cases}
u_t + u_x = 0, & -1 < x < 1, t > 0 \\
u(x,0) = u_0(x), & -1 \le x \le 1 \\
u(-1,t) = g(t), & t > 0.
\end{cases} \tag{2.3}$$

For two constants $\delta_1, \delta_2 \in [0, 1)$, we use a uniform mesh with the mesh size $\Delta x = \frac{2}{N+\delta_1+\delta_2}$ and distribute the mesh points as

$$-1 + \delta_1 \Delta x = x_0 < x_1 < \cdots < x_N = 1 - \delta_2 \Delta x. \tag{2.4}$$

Notice that we have deliberately allowed the physical boundary $x = \pm 1$ not coinciding with grid points. We take $\{x_0, x_1, \cdots, x_N\}$ as our interior points.

In this paper, for simplicity we use the third order upwind-biased conservative finite difference operator (or the corresponding third order WENO operator [12]) to approximate the first order spatial derivative, unless otherwise stated. The methodology certainly also works for other high order finite difference operators. For $j = 0, 1, \cdots, N$, we have the semi-discrete scheme

$$(u_j)_t + \frac{1}{\Delta x}(\hat{u}_{j+\frac{1}{2}} - \hat{u}_{j-\frac{1}{2}}) = 0, \tag{2.5}$$

with the numerical flux $\hat{u}_{j+\frac{1}{2}}$ defined as

$$\hat{u}_{j+\frac{1}{2}} = -\frac{1}{6}u_{j-1} + \frac{5}{6}u_j + \frac{1}{3}u_{j+1}. \tag{2.6}$$

Here u_j is the numerical approximation to the exact solution u at the grid point (x_j, t). The scheme (2.5) requires a four point stencil so up to two ghost points are needed near the boundaries. We concentrate on describing how to define the values at the ghost points $\{x_{-2}, x_{-1}, x_{N+1}\}$.

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

$$u_j = \sum_{k=0}^{s} \frac{(x_j + 1)^k}{k!} \partial_x^{(k)} u, \qquad j = -2, -1, \tag{2.7a}$$

$$u_j = \sum_{k=0}^{s} \frac{(x_j - 1)^k}{k!} \partial_x^{(k)} u, \qquad j = N + 1. \tag{2.7b}$$

Here $\partial_x^{(k)} u$ is the numerical approximation of the k-th spatial derivative of u at the physical boundary $x = -1$ in (2.7a) or at $x = 1$ in (2.7b). For the left boundary, we can

use the equation and the boundary condition to get

$$\begin{aligned}
&\partial_x^{(0)} u = u(-1,t) = g(t),\\
&\partial_x^{(1)} u = u_x(-1,t) = -u_t(-1,t) = -g'(t),\\
&\partial_x^{(2)} u = u_{xx}(-1,t) = -u_{tx}(-1,t) = u_{tt}(-1,t) = g''(t),\\
&\quad \cdots
\end{aligned}$$

Thus we can obtain $\{\partial_x^{(k)} u, k = 0, 1, \cdots\}$ completely from the given boundary condition and the partial differential equation (PDE) by converting the spatial derivative into the time derivatives. This is very similar to the traditional Lax-Wendroff scheme, in which the time derivatives are rewritten in terms of spatial derivatives through repeatedly using the PDE. Here the roles of time and space are reversed, hence the method was referred to as the inverse Lax-Wendroff procedure. In the following, we use the superscript "ilw" to denote derivatives obtained through the ILW procedure, e.g. u_x^{ilw} means that u_x is derived from the ILW procedure. Even though this ILW procedure works for general inflow case for nonlinear problems, multi-dimensions and systems, the algebra could be quite complicated, especially for higher order spatial derivatives. Moreover, this procedure certainly does not apply to the outflow boundary, where no physical boundary condition is provided.

Another way to obtain approximation to these spatial derivatives at the boundary is through the traditional Lagrangian or WENO extrapolation [28, 31] from interior points with suitable order of accuracy. We use the superscript "ext" to denote derivatives obtained through the extrapolation procedure. Extrapolation is needed not only at the outflow boundary, but also at the inflow boundary for as many higher order spatial derivative terms as stability allows. In [31], a simplified ILW procedure is introduced to use the ILW procedure only to obtain the solution and first order spatial derivative at the inflow boundary, and use extrapolation to obtain all other derivatives. This gives a stable fifth order WENO algorithm when the physical point is aligned with the grid.

In [32, 17], systematic analyses are performed for central compact schemes and upwind-biased finite difference schemes respectively for the simpled ILW procedure, to determine the minimum number of spatial derivatives needed from the ILW procedure in order to guarantee stability, regardless of the distance from the physical boundary to the closest grid point.

2.2 One-dimensional scalar convection-diffusion equations

We consider the following one-dimensional scalar convection-diffusion equation

$$\begin{cases} u_t + f(u)_x = \varepsilon u_{xx}, \quad -1 < x < 1, \quad t > 0, \\ u(-1,t) = g_1(t), \quad u(1,t) = g_2(t), \quad t > 0 \\ u(x,0) = u_0(x), \quad -1 \le x \le 1 \end{cases} \tag{2.8}$$

where $\varepsilon > 0$ is a constant. Without loss of generality, we assume $f'(g_1(t)) > 0$ and $f'(g_2(t)) \ge 0$.

We still use the uniform mesh defined in (2.4). The first order spatial derivative is approximated by the third order upwind-biased finite difference operator (or its WENO extension), and the second order spatial derivative is approximated by the central fourth order discretization. For $j = 0, 1, \cdots, N$, we have the semi-discrete scheme

$$(u_j)_t + \frac{1}{\Delta x}(\hat{f}_{j+1/2} - \hat{f}_{j-1/2}) = \frac{\varepsilon}{12\Delta x^2}(-u_{j+2} + 16u_{j+1} - 30u_j + 16u_{j-1} - u_{j-2}) \tag{2.9}$$

where $\hat{f}_{j+1/2}$ is the upwind-biased numerical flux. For instance, if $f'(u) \ge 0$, we have

$$\hat{f}_{j+\frac{1}{2}} = -\frac{1}{6}f(u_{j-1}) + \frac{5}{6}f(u_j) + \frac{1}{3}f(u_{j+1})$$

Our goal is to define the values at the ghost points $\{x_{-2}, x_{-1}, x_{N+1}, x_{N+2}\}$.

We first consider the left boundary $x = -1$. We can always use the given boundary condition to get $u = g_1(t)$ in the Taylor expansion, hence the problem is to define the first and higher order derivatives in the Taylor expansion. As explained before, we have two choices. One is to extrapolate from the interior points; the other is to use the ILW method to get the derivatives from the equation and the given boundary condition.

We start with two extreme cases: (1) $f'(g_1(t)) = 0$, i.e., pure diffusion, and (2) $\varepsilon = 0$, i.e., pure convection. For the pure diffusion case with $f'(g_1(t)) = 0$, we can only obtain the second order (or higher even order) spatial derivative $u_{xx}^{ilw} = \frac{g_1'(t)}{\varepsilon}$ using the ILW method. The first order (or higher odd order) spatial derivative has to come from extrapolation. For the purely convection case with $\varepsilon = 0$, we can obtain the first order (or any higher order) spatial derivative $u_x^{ilw} = \frac{g_1'(t)}{f'(g_1(t))}$ by the ILW method, as in the hyperbolic case. To save the cost, the second order spatial derivative is obtained using extrapolation according to the result in [31], as explained before.

In the general case when $f'(g_1(t)) \neq 0$ and $\varepsilon \neq 0$, we have two ways to perform the ILW procedure. We may obtain the second order spatial derivative by extrapolation, then obtain the first order spatial derivative through the ILW procedure

$$u_x^{ilw} = \frac{1}{f'(g_1(t))}(-g_1'(t) + \varepsilon u_{xx}^{ext}),$$

which is what we would do for the purely convection case and thus is expected to work in the convection-dominated regime. Alternately, we may obtain the first order spatial derivative by extrapolation, then obtain the second order spatial derivative through the ILW procedure

$$u_{xx}^{ilw} = \frac{1}{\varepsilon}(g_1'(t) + f'(g_1(t))u_x^{ext}),$$

which is what we would do for the purely diffusion case and therefore is expected to work in the diffusion-dominated regime.

In order to obtain a strategy which is applicable for all convection-diffusion equations, we form a convex combination of the two procedures described above

$$u_x = \alpha u_x^{ilw} + (1-\alpha)u_x^{ext}, \quad u_{xx} = (1-\alpha)u_{xx}^{ilw} + \alpha u_{xx}^{ext},$$

where $0 \leq \alpha \leq 1$ satisfies the following conditions: $\alpha = 0$ for the pure diffusion case with $f'(g_1(t)) = 0$; and $\alpha = 1$ for the pure convection case with $\varepsilon = 0$. After extensive numerical tests, we have settled on the choice

$$\alpha = \frac{f'(g_1(t))^2 \Delta x^2}{f'(g_1(t))^2 \Delta x^2 + 9\varepsilon^2}. \tag{2.10}$$

Extending the strategy to all the spatial derivatives, we have

$$\begin{pmatrix} \partial_x^{(0)} u \\ \partial_x^{(1)} u \\ \partial_x^{(2)} u \end{pmatrix} = \frac{f'(g_1(t))^2 \Delta x^2}{f'(g_1(t))^2 \Delta x^2 + 9\varepsilon^2} \begin{pmatrix} g_1(t) \\ u_x^{ilw} \\ u_{xx}^{ext} \end{pmatrix} + \frac{9\varepsilon^2}{f'(g_1(t))^2 \Delta x^2 + 9\varepsilon^2} \begin{pmatrix} g_1(t) \\ u_x^{ext} \\ u_{xx}^{ilw} \end{pmatrix}. \tag{2.11}$$

Using these values, we can obtain the values of the numerical solution at the ghost points near the left boundary by the Taylor expansion (2.7a). The same procedure can be used on the right boundary $x = 1$.

2.3 One-dimensional linear systems

We now extend the numerical boundary condition to one-dimensional linear systems. We consider the following equations

$$\boldsymbol{U}_t + A\boldsymbol{U}_x = B\boldsymbol{U}_{xx}, \qquad -1 < x < 1, t > 0 \tag{2.12}$$

with appropriate initial and boundary conditions. Here

$$\boldsymbol{U} = (U_1, U_2, \cdots, U_d)^T,$$

A and B are constant $d \times d$ matrices. The eigenvalues of A are real and A has a complete set of eigenvectors. B is a positive-definite matrix. We still have the uniform mesh defined in (2.4). Our goal is to define the values at the ghost points $\{x_{-2}, x_{-1}, x_{N+1}, x_{N+2}\}$.

The main complication for the system case is that the convection term and the diffusion term in general cannot be simultaneously diagonalized. This makes a direct application of the strategy for the scalar convection-diffusion equation difficult. Without loss of generality, we only consider the left boundary.

2.3.1 Convection-dominated case

In the convection-dominated case, we first diagonalize the matrix A and obtain its eigenvalues $\{\lambda_1, \lambda_2, \cdots, \lambda_d\}$ and the corresponding left eigenvectors $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \cdots, \boldsymbol{l}_d\}$ to form the matrix

$$L = \begin{pmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \cdots \\ \boldsymbol{l}_d \end{pmatrix} = \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,d} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,d} \\ \cdots & \cdots & \cdots & \cdots \\ l_{d,1} & l_{d,2} & \cdots & l_{d,d} \end{pmatrix}.$$

Then we have $LAL^{-1} = \Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_d)$. With this characteristic decomposition $\boldsymbol{V} = L\boldsymbol{U}$, the equations (2.12) becomes

$$\boldsymbol{V}_t + \Lambda \boldsymbol{V}_x = LB\boldsymbol{U}_{xx} = \boldsymbol{RHS}^{\boldsymbol{V}}. \tag{2.13}$$

Since B is a positive-definite matrix, we should have d boundary conditions at the boundary for well-posedness. However, in the convection-dominated case, we impose characteristic boundary conditions for $\boldsymbol{V}$, depending on the signs of the eigenvalues in Λ.

We look at the specific case of $\{\lambda_j \leq 0, j = 1, 2, \cdots, k_n\}$ and $\{\lambda_j > 0, j = k_n + 1, k_n + 2, \cdots, d\}$. The characteristic variables $\{V_j, j = 1, \cdots, k_n\}$ have outflow boundary condition at $x = -1$, and $\{V_j, j = k_n + 1, \cdots, d\}$ have inflow boundary conditions at $x = -1$. We denote $\partial_x^{(k)} \boldsymbol{U}_c$ as the numerical approximation to $\frac{\partial^k \boldsymbol{U}}{\partial x^k}$ on the boundary. The subscript "c" indicates the convection-dominated case. For $j = 1, \cdots, k_n$, the outgoing characteristic variable V_j can be extrapolated from interior point values $\{(V_j)_i, i = 0, 1, 2, 3\}$ to the boundary to get $\{(V_j)^{ext}, (V_j)_x^{ext}, (V_j)_{xx}^{ext}\}$. For $j = k_n + 1, \cdots, d$, the incoming characteristic variable is set by the given boundary condition as $V_j = (L\boldsymbol{g})_j(t)$, where $\boldsymbol{g}(t)$ is the given boundary condition for $\boldsymbol{U}$ and $(L\boldsymbol{g})_j(t)$ is the j-th entry of the vector $L\boldsymbol{g}$. With the extrapolation values and boundary conditions we have

$$\begin{cases} \boldsymbol{l}_j \cdot \partial_x^{(0)} \boldsymbol{U}_c = (V_j)^{ext}, & j = 1, \cdots, k_n \\ \boldsymbol{l}_j \cdot \partial_x^{(0)} \boldsymbol{U}_c = (L\boldsymbol{g})_j(t), & j = k_n + 1, \cdots, d \end{cases} \tag{2.14}$$

Next we would like to find the first order derivatives $\partial_x^{(1)} \boldsymbol{U}$ by the inverse Lax-Wendroff procedure, which is the essential part of the algorithm in [28, 31]. Since the boundary conditions for $\{V_j, j = k_n + 1, \cdots, d\}$ are prescribed, combining with the information of outgoing characteristic variable $\{V_j, j = 1, \cdots, k_n\}$, we have

$$\begin{cases} \boldsymbol{l}_j \cdot \partial_x^{(1)} \boldsymbol{U}_c = (V_j)_x^{ext}, & j = 1, \cdots, k_n \\ (LA)_j \cdot \partial_x^{(1)} \boldsymbol{U}_c = -(L\boldsymbol{g})_j'(t) + RHS_j^{\boldsymbol{V}}, & j = k_n + 1, \cdots, d \end{cases} \tag{2.15}$$

where $(LA)_m$ is the m-th row vector of the matrix LA and $RHS_m^{\boldsymbol{V}}$ is the m-th entry of the vector $\boldsymbol{RHS}^{\boldsymbol{V}}$ in (2.13). All second order derivatives $\boldsymbol{U}_{xx}$ in $\boldsymbol{RHS}^{\boldsymbol{V}}$ are obtained by

extrapolation. After we obtain $\partial_x^{(1)}\boldsymbol{U}_c$, the second order derivatives $\partial_x^{(2)}\boldsymbol{U}_c$ are obtained by extrapolation.

2.3.2 Diffusion-dominated case

We now turn to the diffusion-dominated case. Firstly we diagonalize the matrix B and obtain its eigenvalues $\{\lambda'_1, \lambda'_2, \cdots, \lambda'_d\}$ and the corresponding left eigenvectors $\{\boldsymbol{l}'_1, \boldsymbol{l}'_2, \cdots, \boldsymbol{l}'_d\}$ to form the matrix

$$L' = \begin{pmatrix} \boldsymbol{l}'_1 \\ \boldsymbol{l}'_2 \\ \cdots \\ \boldsymbol{l}'_d \end{pmatrix} = \begin{pmatrix} l'_{1,1} & l'_{1,2} & \cdots & l'_{1,d} \\ l'_{2,1} & l'_{2,2} & \cdots & l'_{2,d} \\ \cdot & \cdot & \cdots & \cdot \\ l'_{d,1} & l'_{d,2} & \cdots & l'_{d,d} \end{pmatrix}$$

Then we have $L'BL'^{-1} = \Lambda' = diag(\lambda'_1, \lambda'_2, \cdots, \lambda'_d)$. Since the matrix B is positive-definite, the eigenvalues are positive. With this characteristic decomposition $\boldsymbol{W} = L'\boldsymbol{U}$, the equations (2.12) becomes

$$\boldsymbol{W}_t + C\boldsymbol{W}_x = \Lambda'\boldsymbol{W}_{xx}, \tag{2.16}$$

where $C = L'AL'^{-1}$. Our algorithm to get the derivatives of each order is as follows

$$\partial_x^{(0)}\boldsymbol{U}_d = \boldsymbol{g}(t), \tag{2.17}$$

$$\partial_x^{(1)}\boldsymbol{U}_d = \boldsymbol{U}_x^{ext}, \tag{2.18}$$

$$\partial_x^{(2)}\boldsymbol{U}_d = B^{-1}(-\boldsymbol{g}'(t) + A\boldsymbol{U}_x^{ext}), \tag{2.19}$$

where $\partial_x^{(k)}\boldsymbol{U}_d$ is the numerical approximation to $\frac{\partial^k \boldsymbol{U}}{\partial x^k}$ in the diffusion-dominated case.

2.3.3 General convection-diffusion case

Now we combine the convection- and diffusion-dominated cases together. In order to obtain the convex combination coefficient, we would like to use the matrices C and Λ' in (2.16). First we define the coefficients for both the convection- and diffusion-dominated cases as follows

$$a_j = \Delta x^2 \sum_{i=1}^{d} C_{j,i}^2, \quad \varepsilon_j = 9\lambda_j'^2, \quad \alpha_j = \frac{a_j}{a_j + \varepsilon_j}, \quad j = 1, 2, \cdots, d, \tag{2.20}$$

where $C_{j,i}$ is the (j,i) entry of matrix C in (2.16). We then consider equation (2.16), and take a transformation of $\partial_x^{(k)}\boldsymbol{U}_c$ and $\partial_x^{(k)}\boldsymbol{U}_d$ as

$$\partial_x^{(k)}\boldsymbol{W}_c = L'\partial_x^{(k)}\boldsymbol{U}_c, \qquad \partial_x^{(k)}\boldsymbol{W}_d = L'\partial_x^{(k)}\boldsymbol{U}_d, \qquad k = 0, 1, 2, \tag{2.21}$$

and make a convex combination of them

$$\partial_x^{(k)}W_j = \alpha_j\partial_x^{(k)}(W_c)_j + (1-\alpha_j)\partial_x^{(k)}(W_d)_j, \quad k = 0, 1, 2. \tag{2.22}$$

After we get $\{\partial_x^{(k)}\boldsymbol{W}, k = 0, 1, 2\}$, we transform back and obtain

$$\partial_x^{(k)}\boldsymbol{U} = L'^{-1}\partial_x^{(k)}\boldsymbol{W}, \quad k = 0, 1, 2. \tag{2.23}$$

With these derivatives, we can use Taylor expansion to get values of the numerical solution at the ghost points.

We summarize our algorithm of the boundary treatment as follows, under the assumption that $\{\boldsymbol{U}_j, j = 0, 1, \cdots, N\}$ have been updated at time level t_n. Here we just take the left boundary as an example to illustrate our algorithm.

Algorithm 1:

1. For the convection-dominated case:

   - Compute the eigenvalues of A to decide the inflow and outflow boundary conditions. Use the characteristic decomposition of A and obtain the equation (2.13).
   - For each outgoing characteristic variable V_j, we use extrapolation to obtain $\{(V_j)^{ext}, (V_j)_x^{ext}, (V_j)_{xx}^{ext}\}$. With the prescribed boundary conditions for the incoming characteristic variables and extrapolated values $(V_j)^{ext}$, we can obtain $\partial_x^{(0)}\boldsymbol{U}_c$ by solving the linear system (2.14).
   - With the inverse Lax-Wendroff procedure for the incoming characteristic variables and extrapolated values $(V_j)_x^{ext}$, we can obtain $\partial_x^{(1)}\boldsymbol{U}_c$ by solving the linear system (2.15).

- The second order derivative is obtained by extrapolation, i.e. $\partial_x^{(2)}\boldsymbol{U}_c = \boldsymbol{U}_{xx}^{ext}$.

2. For the diffusion-dominated case:

   - Diagonalize the matrix B and obtain the equation (2.16).
   - Impose boundary conditions for $\partial_x^{(0)}\boldsymbol{U}_d$ on the boundary, as in (2.17).
   - The first order derivative $\partial_x^{(1)}\boldsymbol{U}_d$ is obtained by extrapolation, as in (2.18).
   - Use the inverse Lax-Wendroff method to obtain the second order derivative $\partial_x^{(2)}\boldsymbol{U}_d$, as in (2.19).

3. For the general convection-diffusion case:

   - Choose the coefficient $\{\alpha_j, j = 1, \cdots, d\}$ defined in (2.20).
   - Convert $\partial_x^{(k)}\boldsymbol{U}_c$ and $\partial_x^{(k)}\boldsymbol{U}_d$ into $\partial_x^{(k)}\boldsymbol{W}_c$ and $\partial_x^{(k)}\boldsymbol{W}_d$, $k = 0, 1, 2$ respectively, as in (2.21).
   - Make a convex combination of the derivatives $\partial_x^{(k)}\boldsymbol{W}_c$ and $\partial_x^{(k)}\boldsymbol{W}_d$ to get the derivatives $\partial_x^{(k)}\boldsymbol{W}$, $k = 0, 1, 2$, as in (2.22).
   - Transform from $\partial_x^{(k)}\boldsymbol{W}$ back to $\partial_x^{(k)}\boldsymbol{U}$, $k = 0, 1, 2$, as in (2.23).

4. Assign the values of the ghost points by the Taylor expansion (2.7).

2.4 One-dimensional Navier-Stokes equations

We consider the one-dimensional compressible Navier-Stokes equations in the non-dimensional form [33]

$$\begin{cases} \rho_t + (\rho u)_x = 0 \\ (\rho u)_t + (\rho u^2 + p)_x = \frac{1}{Re}\left(\frac{4}{3}u\right)_{xx} \\ E_t + (u(E+p))_x = \frac{1}{Re}\left[\frac{2}{3}(u^2)_{xx} + \frac{1}{(\gamma-1)Pr}(c^2)_{xx}\right] \end{cases} \tag{2.24}$$

with appropriate initial and boundary conditions. Here ρ is the density, u is the velocity, E is the total energy, p is the pressure, the equation of state is

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2,$$

with $\gamma = 1.4$ for air. The speed of sound is given as

$$c = \sqrt{\frac{\gamma p}{\rho}}.$$

Re is the Reynolds number, and $Pr = 0.7$ is the Prandtl number.

Compared with one-dimensional linear system, there are two more features of the Navier-Stokes equations. The first one is nonlinearity. The second one is that there is no diffusion term in the density equation. In order to motivate our choice of the numerical boundary condition, we rewrite the system into the following form for smooth solutions

$$\boldsymbol{U}_t + A(\boldsymbol{U})\boldsymbol{U}_x = B(\boldsymbol{U})\boldsymbol{U}_{xx} + \boldsymbol{NLT}, \tag{2.25}$$

where

$$\boldsymbol{U} = (U_1, U_2, U_3)^T = (\rho, \rho u, E)^T,$$

$$A(\boldsymbol{U}) = \begin{pmatrix} 0 & 1 & 0 \\ \frac{1}{2}(\gamma-3)u^2 & (3-\gamma)u & \gamma - 1 \\ \frac{1}{2}(\gamma-1)u^3 - uH & H - (\gamma-1)u^2 & \gamma u \end{pmatrix},$$

$$B(\boldsymbol{U}) = \frac{1}{Re}\begin{pmatrix} 0 & 0 & 0 \\ -\frac{4u}{3\rho} & \frac{4}{3\rho} & 0 \\ \left(\frac{\gamma}{Pr} - \frac{4}{3}\right)\frac{u^2}{\rho} - \frac{\gamma}{Pr}\frac{E}{\rho^2} & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)\frac{u}{\rho} & \frac{\gamma}{Pr\rho} \end{pmatrix},$$

and the additional nonlinear term is given by

$$\boldsymbol{NLT} = \frac{1}{Re}\begin{pmatrix} 0 \\ \frac{8}{3\rho^3}\Big(-\rho\rho_x(\rho u)_x + \rho_x^2 \rho u\Big) \\ \frac{1}{\rho^2}\Big(\frac{4}{3} - \frac{\gamma}{Pr}\Big)(3u\rho_x - (\rho u)_x)(u\rho_x - (\rho u)_x) + \frac{2\gamma}{Pr\rho^3}\Big(-\rho\rho_x E_x + \rho_x^2 E\Big) \end{pmatrix}.$$

Here, H is enthalpy defined by

$$H = \frac{E + p}{\rho}.$$

2.4.1 Convection-dominated case

We first consider the convection-dominated case, i.e., when the Reynolds number Re is large. We treat this case along the lines described in the previous subsection and in

the same way as in [31] for the one-dimensional Euler equations. Namely, we apply the inverse Lax-Wendroff procedure for the first order derivative terms, and use extrapolation to obtain all the second order derivatives terms, including those in the nonlinear term.

Considering the left boundary $x = -1$, we first determine the boundary value $\boldsymbol{U}_b$ through the given boundary condition and extrapolation for the variables not given by the boundary condition. We then use this boundary value to diagonalize the convection term $A(\boldsymbol{U}_b)$ and obtain its three eigenvalues $\lambda_1 = u - c, \lambda_2 = u, \lambda_3 = u + c$ and the corresponding left eigenvectors $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3\}$ which form the matrix

$$L = \begin{pmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \boldsymbol{l}_3 \end{pmatrix} = \begin{pmatrix} l_{1,1} & l_{1,2} & l_{1,3} \\ l_{2,1} & l_{2,2} & l_{2,3} \\ l_{3,1} & l_{3,2} & l_{3,3} \end{pmatrix}.$$

Thus we have $LAL^{-1} = \Lambda = diag(\lambda_1, \lambda_2, \lambda_3)$ on the left boundary $x = -1$. With this characteristic decomposition $\boldsymbol{V} = L\boldsymbol{U}$, the equations (2.25) on the left boundary becomes

$$\boldsymbol{V}_t + \Lambda \boldsymbol{V}_x = L(B(\boldsymbol{U})\boldsymbol{U}_{\boldsymbol{xx}} + \boldsymbol{NLT}) = \boldsymbol{RHS}^{\boldsymbol{V}}. \tag{2.26}$$

The number of given boundary conditions depends on the number of positive eigenvalues in Λ. This is the same as in Euler equations, although the number of given boundary conditions for the Navier-Stokes equations is always equal to or more than that for the Euler equations. In the case that the number of given boundary conditions for the Navier-Stokes equations is more than that for the Euler equations, we take the given boundary condition for the convection-dominated case sequentially by the following ordered list: E, ρu and ρ, until the required number is reached.

We now look at the specific case of $\lambda_1 \leq 0$, $\lambda_2 > 0$, $\lambda_3 > 0$. The characteristic variable V_1 has an outflow boundary condition at $x = -1$, and V_2, V_3 have inflow boundary conditions at $x = -1$. For simplicity, we denote $\partial_x^{(k)} \boldsymbol{U}_c$ as the numerical approximation to $\frac{\partial^k \boldsymbol{U}}{\partial x^k}$ at the boundary. By the ordered preference list above, we should impose $\partial_x^{(0)}(U_c)_2 = g_2(t)$ and $\partial_x^{(0)}(U_c)_3 = g_3(t)$ using the given boundary conditions. For the outgoing characteristic variable V_1, we extrapolate from $\{(V_1)_i, i = 0, 1, 2, 3\}$ to the

boundary to get $\{(V_1)^{ext}, (V_1)_x^{ext}, (V_1)_{xx}^{ext}\}$. Finally, $\partial_x^{(0)}(U_c)_1$ can be obtained from the following equations

$$\begin{cases} \partial_x^{(0)}(U_c)_2 = g_2(t), \\ \partial_x^{(0)}(U_c)_3 = g_3(t), \\ \boldsymbol{l}_1 \cdot \partial_x^{(0)} \boldsymbol{U}_c = (V_1)^{ext}. \end{cases} \tag{2.27}$$

Next we would like to find the first order derivatives $\partial_x^{(1)}\boldsymbol{U}_c$ by the inverse Lax-Wendroff procedure. Since the boundary conditions for U_2 and U_3 are prescribed, combining with the information of outgoing characteristic variable V_1, we have

$$\begin{cases} \boldsymbol{l}_1 \cdot \partial_x^{(1)} \boldsymbol{U}_c = (V_1)_x^{ext}, \\ A_2 \cdot \partial_x^{(1)} \boldsymbol{U}_c = -g_2'(t) + RHS_2^{\boldsymbol{V}}, \\ A_3 \cdot \partial_x^{(1)} \boldsymbol{U}_c = -g_3'(t) + RHS_3^{\boldsymbol{V}}, \end{cases} \tag{2.28}$$

where A_m is the m-th row vector of the matrix A and $RHS_m^{\boldsymbol{V}}$ is the m-th entry of the vector $\boldsymbol{RHS}^{\boldsymbol{V}}$ in (2.26), $m = 1, 2, 3$. All derivatives in $\boldsymbol{RHS}^{\boldsymbol{V}}$ are obtained by extrapolation. After we obtain $\partial_x^{(1)}\boldsymbol{U}_c$, the second order derivatives $\partial_x^{(2)}\boldsymbol{U}_c$ are obtained by extrapolation.

2.4.2 Diffusion-dominated case

We now turn to the diffusion-dominated case, i.e., when Re is small. Special attention must be paid to the fact that this is an incompletely parabolic system. In [26, 8], the authors decoupled this incompletely parabolic system into a strictly hyperbolic system and a parabolic system, and then deduced sufficient conditions for the well-posedness of the incompletely parabolic system. We take this decoupling as a guidance for the design of our algorithm in the diffusion-dominated case. We diagonalize the matrix in front of the second order derivative terms on the boundary to obtain

$$\boldsymbol{W}_t + C\boldsymbol{W}_x = \Lambda' \boldsymbol{W}_{xx} + L'\boldsymbol{NLT}, \tag{2.29}$$

where

$$\boldsymbol{W} = L'\boldsymbol{U}, \quad L'BL'^{-1} = \Lambda' = \frac{1}{Re\rho} diag\left(0, \frac{4}{3}, \frac{\gamma}{Pr}\right),$$

and

$$C = \begin{pmatrix} u & -\frac{c}{\gamma} & 0 \\ -c & u & -c \\ 0 & -\frac{\gamma-1}{\gamma}c & u \end{pmatrix}, \quad L' = \begin{pmatrix} \boldsymbol{l}'_1 \\ \boldsymbol{l}'_2 \\ \boldsymbol{l}'_3 \end{pmatrix} = \begin{pmatrix} l'_{1,1} & l'_{1,2} & l'_{1,3} \\ l'_{2,1} & l'_{2,2} & l'_{2,3} \\ l'_{3,1} & l'_{3,2} & l'_{3,3} \end{pmatrix},$$

where $\{\boldsymbol{l}'_1, \boldsymbol{l}'_2, \boldsymbol{l}'_3\}$ forms a complete set of left eigenvectors of the matrix $B(\boldsymbol{U})$ on the boundary. We consider the linear part of (2.29), $\boldsymbol{W}_t + C\boldsymbol{W}_x = \Lambda'\boldsymbol{W}_{xx}$, which is an incomplete parabolic equations. In [8], they separated this linear system into

$$\begin{aligned} &(W_1)_t + C_{1,1}(W_1)_x = 0, \\ &\begin{pmatrix} W_2 \\ W_3 \end{pmatrix}_t + \begin{pmatrix} C_{2,2} & C_{2,3} \\ C_{3,2} & C_{3,3} \end{pmatrix} \begin{pmatrix} W_2 \\ W_3 \end{pmatrix}_x = \frac{1}{Re\rho} \begin{pmatrix} \frac{4}{3} & 0 \\ 0 & \frac{\gamma}{Pr} \end{pmatrix} \begin{pmatrix} W_2 \\ W_3 \end{pmatrix}_{xx}. \end{aligned}$$

This is a combination of a scalar hyperbolic equation and a parabolic system. With suitable initial and boundary conditions, they proved that this system is well-posed. This motivates us to treat the original system as a combination of a hyperbolic equation and a parabolic system. Let us consider two cases: $u > 0$ and $u \leq 0$ on the left boundary $x = -1$. If $u > 0$, then we have three given boundary conditions $\boldsymbol{g}(t)$ on the left boundary. Our algorithm to get the derivatives of each order is as follows

$$\partial_x^{(0)} \boldsymbol{U}_d = \boldsymbol{g}(t), \tag{2.30}$$

$$\begin{cases} A_1 \cdot \partial_x^{(1)} \boldsymbol{U}_d = -g'_1(t) + B_1 \cdot \boldsymbol{U}_{xx}^{ext} + NLT_1, \\ \boldsymbol{l}'_2 \cdot \partial_x^{(1)} \boldsymbol{U}_d = (W_2)_x^{ext}, \\ \boldsymbol{l}'_3 \cdot \partial_x^{(1)} \boldsymbol{U}_d = (W_3)_x^{ext}, \end{cases} \tag{2.31}$$

$$\begin{cases} \boldsymbol{l}'_1 \cdot \partial_x^{(2)} \boldsymbol{U}_d = (W_1)_{xx}^{ext}, \\ B_2 \cdot \partial_x^{(2)} \boldsymbol{U}_d = g'_2(t) + A_2 \cdot \boldsymbol{U}_x^{ext} - NLT_2, \\ B_3 \cdot \partial_x^{(2)} \boldsymbol{U}_d = g'_3(t) + A_3 \cdot \boldsymbol{U}_x^{ext} - NLT_3, \end{cases} \tag{2.32}$$

where B_m is the m-th row vector of the matrix B, $m = 1, 2, 3$, and the derivatives in $\boldsymbol{NLT}$ in (2.31) and (2.32) are obtained by extrapolation.

For $u \leq 0$, we only have two given boundary conditions. Since the second order derivatives appear in the momentum and energy equations in (2.24), we take $\rho u, E$ as the

variables with given boundary conditions, and ρ is obtained by extrapolation. Namely, the equations (2.30) and (2.31) are replaced by (2.33) and (2.34) to obtain $\partial_x^{(0)}\boldsymbol{U}_d$ and $\partial_x^{(1)}\boldsymbol{U}_d$, while (2.32) stays the same.

$$\begin{cases} \partial_x^{(0)}(U_d)_1 = (U_1)^{ext}, \\ \partial_x^{(0)}(U_d)_2 = g_2(t), \\ \partial_x^{(0)}(U_d)_3 = g_3(t), \end{cases} \tag{2.33}$$

$$\partial_x^{(1)}\boldsymbol{U}_d = \boldsymbol{U}_x^{ext}. \tag{2.34}$$

2.4.3 General convection-diffusion case

Now we combine the two cases together. In order to obtain the convex combination coefficient, we would like to use the matrices C and Λ' in (2.29). First we define the coefficients for both the convection-dominated case and the diffusion-dominated case as follows

$$\begin{aligned} a_1 &= (C_{1,1}^2 + C_{1,2}^2 + C_{1,3}^2)\Delta x^2, \quad \varepsilon_1 = \frac{9}{2}(\Lambda_{2,2}'^2 + \Lambda_{3,3}'^2), \quad \alpha_1 = \frac{a_1}{a_1 + \varepsilon_1}, \\ a_2 &= (C_{2,1}^2 + C_{2,2}^2 + C_{2,3}^2)\Delta x^2, \quad \varepsilon_2 = 9\Lambda_{2,2}'^2, \quad \alpha_2 = \frac{a_2}{a_2 + \varepsilon_2}, \\ a_3 &= (C_{3,1}^2 + C_{3,2}^2 + C_{3,3}^2)\Delta x^2, \quad \varepsilon_3 = 9\Lambda_{3,3}'^2, \quad \alpha_3 = \frac{a_3}{a_3 + \varepsilon_3}. \end{aligned} \tag{2.35}$$

We then consider equation (2.29), and take a transformation of $\partial_x^{(k)}\boldsymbol{U}_c$ and $\partial_x^{(k)}\boldsymbol{U}_d$, $k = 0, 1, 2$ as

$$\partial_x^{(k)}\boldsymbol{W}_c = L'\partial_x^{(k)}\boldsymbol{U}_c, \qquad \partial_x^{(k)}\boldsymbol{W}_d = L'\partial_x^{(k)}\boldsymbol{U}_d, \qquad k = 0, 1, 2 \tag{2.36}$$

and make a convex combination of them

$$\partial_x^{(k)}W_i = \alpha_i\partial_x^{(k)}(W_c)_i + (1 - \alpha_i)\partial_x^{(k)}(W_d)_i, \quad i = 1, 2, 3, \quad k = 0, 1, 2. \tag{2.37}$$

Notice that we have taken ε_1 as the average of ε_2 and ε_3 because $\Lambda_{1,1}'$ is always 0. If we took $\varepsilon_1 = 9\Lambda_{1,1}'^2$, we would not be able to recover the algorithm for the diffusion-dominated case because $\partial_x^{(k)}W_1$ would always equal to $\partial_x^{(k)}(W_c)_1$. After we obtain $\partial_x^{(k)}\boldsymbol{W}, k = 0, 1, 2$, we transform back and have

$$\partial_x^{(k)}\boldsymbol{U} = L'^{-1}\partial_x^{(k)}\boldsymbol{W}, \quad k = 0, 1, 2. \tag{2.38}$$

With these derivatives, we can use Taylor expansion (2.7a) to impose values of the numerical solution at the ghost points.

Let us summarize our algorithm of the boundary treatment for the Navier-Stokes equations as follows, assuming that $\{\boldsymbol{U}_j, j = 0, 1, \cdots, N\}$ have been updated at time level t_n. We only consider the left boundary here, since there is no difference in applying this method to the right boundary.

Algorithm 2:

1. For the convection-dominated case:

   - Compute the eigenvalues of A in (2.25) to decide the inflow and outflow boundary conditions. Use the characteristic decomposition of A and obtain the equation (2.26).
   - For each outgoing characteristic variable V_j, we use extrapolation to obtain $\{(V_j)^{ext}, (V_j)_x^{ext}, (V_j)_{xx}^{ext}\}$. With the prescribed boundary conditions for the incoming characteristic variables and extrapolated values $(V_j)^{ext}$, we can obtain $\partial_x^{(0)}\boldsymbol{U}_c$ by solving a linear system, such as (2.27).
   - With the inverse Lax-Wendroff procedure for the incoming characteristic variables and extrapolated values $(V_j)_x^{ext}$, we can obtain $\partial_x^{(1)}\boldsymbol{U}_c$ by solving a linear system, such as (2.28).
   - The second order derivatives are obtained by extrapolation, i.e. $\partial_x^{(2)}\boldsymbol{U}_c = \boldsymbol{U}_{xx}^{ext}$.

2. In the diffusion-dominated case, we have two different cases depending on the sign of u. Here we consider $u > 0$.

   - We have three boundary conditions for $\partial_x^{(0)}\boldsymbol{U}_d$, as in (2.30).

- The first order derivative $\partial_x^{(1)}\boldsymbol{U}_d$ are obtained by applying the inverse Lax-Wendroff method to the mass equation, together with the extrapolation equations for $\partial_x^{(1)}(W_d)_2$ and $\partial_x^{(1)}(W_d)_3$, as in (2.31).
- The second order derivative $\partial_x^{(2)}\boldsymbol{W}_d$ are obtained by applying the inverse Lax-Wendroff method to momentum equation and energy equation, together with the extrapolation equation for $\partial_x^{(2)}(W_d)_1$, as in (2.32).

3. For the general case, to combine $\partial_x^{(k)}\boldsymbol{U}_c$ and $\partial_x^{(k)}\boldsymbol{U}_d$, $k = 0, 1, 2$:
   - Choose the coefficient $\{\alpha_j, j = 1, 2, 3\}$ defined in (2.35).
   - Convert $\partial_x^{(k)}\boldsymbol{U}_c$ and $\partial_x^{(k)}\boldsymbol{U}_d$ into $\partial_x^{(k)}\boldsymbol{W}_c$ and $\partial_x^{(k)}\boldsymbol{W}_d$, $k = 0, 1, 2$, respectively, as in (2.36).
   - Make a convex combination of the derivatives $\partial_x^{(k)}\boldsymbol{W}_c$ and $\partial_x^{(k)}\boldsymbol{W}_d$ to obtain the derivatives $\partial_x^{(k)}\boldsymbol{W}$, $k = 0, 1, 2$, as in (2.37).
   - Transform from $\partial_x^{(k)}\boldsymbol{W}$ back to $\partial_x^{(k)}\boldsymbol{U}$, $k = 0, 1, 2$, as in (2.38).
4. Assign the values of the ghost points by the Taylor expansion (2.7a).

Remark: If $u \leq 0$, we replace (2.30) and (2.31) by (2.33) and (2.34) to obtain $\partial_x^{(0)}\boldsymbol{U}_d$ and $\partial_x^{(1)}\boldsymbol{U}_d$ in Step 2, while other steps remain the same.

2.5 Two-dimensional Navier-Stokes equations

We consider the two-dimensional non-dimensionalized compressible Navier-Stokes equations (see e.g. [27, 33])

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y = \frac{1}{Re}(\boldsymbol{S_1}(\boldsymbol{U})_x + \boldsymbol{S_2}(\boldsymbol{U})_y), \quad (x, y) \in \Omega,\ t > 0, \tag{2.39}$$

with suitable initial and boundary conditions, where

$$\boldsymbol{U} = (U_1, U_2, U_3, U_4)^T = (\rho, \rho u, \rho v, E)^T,$$

$$\boldsymbol{F}(\boldsymbol{U}) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))^T,$$
$$\boldsymbol{G}(\boldsymbol{U}) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))^T,$$
$$\boldsymbol{S_1}(\boldsymbol{U}) = (0, \tau_{11}, \tau_{21}, \sigma_1)^T,$$
$$\boldsymbol{S_2}(\boldsymbol{U}) = (0, \tau_{12}, \tau_{22}, \sigma_2)^T,$$

with the components of the viscous stress tensor given by

$$\tau_{11} = \frac{4}{3}u_x - \frac{2}{3}v_y, \quad \tau_{12} = \tau_{21} = u_y + v_x, \quad \tau_{22} = \frac{4}{3}v_y - \frac{2}{3}u_x.$$

and

$$\sigma_1 = u\tau_{11} + v\tau_{12} + \frac{(c^2)_x}{(\gamma-1)Pr}, \quad \sigma_2 = u\tau_{21} + v\tau_{22} + \frac{(c^2)_y}{(\gamma-1)Pr}.$$

As in the one-dimensional case, Re is the Reynolds number, $\gamma = 1.4$ for air, and $Pr = 0.7$ is the Prandtl number. We also have

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

To define the value at a ghost point $P = (x_i, y_j)$, we find a point $P_0 = (x_0, y_0)$ on the physical boundary $\partial\Omega$ such that the outward normal vector $\boldsymbol{n}$ at P_0 goes through the ghost point P. For the ease of the inverse Lax-Wendroff method, we set up a local coordinate system at P_0 by

$$\begin{pmatrix}\hat{x}\\ \hat{y}\end{pmatrix} = T\begin{pmatrix}x\\ y\end{pmatrix}, \quad T = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}, \tag{2.40}$$

where θ is the angle between $\boldsymbol{n}$ and the x-axis and T is the rotation matrix. After this transformation we have $\hat{x}$ sharing the same direction with the normal $\boldsymbol{n}$ and $\hat{y}$ becomes the tangential direction of $\partial\Omega$ at P_0. If we take

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

then we get a new system

$$\hat{\boldsymbol{U}}_t + \boldsymbol{F}(\hat{\boldsymbol{U}})_{\hat{x}} + \boldsymbol{G}(\hat{\boldsymbol{U}})_{\hat{y}} = \frac{1}{Re}(\boldsymbol{S_1}(\hat{\boldsymbol{U}})_{\hat{x}} + \boldsymbol{S_2}(\hat{\boldsymbol{U}})_{\hat{y}}) \tag{2.41}$$

with the new variables

$$\hat{\boldsymbol{U}} = (\hat{U}_1, \hat{U}_2, \hat{U}_3, \hat{U}_4)^T = (\rho, \rho\hat{u}, \rho\hat{v}, E)^T. \tag{2.42}$$

We concentrate on this system, which is the same as the original Navier-Stokes system (2.39) because of rotational invariance. Our goal is to obtain $\{\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}},\, k = 0, 1, 2\}$, which is a numerical approximation to $\frac{\partial^k \hat{\boldsymbol{U}}}{\partial \hat{x}^k}$ at P_0. First we rewrite the system (2.41) as

$$\hat{\boldsymbol{U}}_t + A(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{x}} + B(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{y}} = C(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{x}\hat{x}} + D(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{y}\hat{y}} + E(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{x}\hat{y}} + \boldsymbol{NLT} \tag{2.43}$$

where

$$A(\hat{\boldsymbol{U}}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ (\gamma-1)q - \hat{u}^2 & (3-\gamma)\hat{u} & (1-\gamma)\hat{v} & \gamma - 1 \\ -\hat{u}\hat{v} & \hat{v} & \hat{u} & 0 \\ -\hat{u}H + (\gamma-1)\hat{u}q & H + (1-\gamma)\hat{u}^2 & (1-\gamma)\hat{u}\hat{v} & \gamma\hat{u} \end{pmatrix}$$

with

$$q = \frac{1}{2}(\hat{u}^2 + \hat{v}^2)$$

We also have

$$B(\hat{\boldsymbol{U}}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -\hat{u}\hat{v} & \hat{v} & \hat{u} & 0 \\ (\gamma-1)q - \hat{v}^2 & (1-\gamma)\hat{u} & (3-\gamma)\hat{v} & \gamma - 1 \\ -\hat{v}H + (\gamma-1)\hat{v}q & (1-\gamma)\hat{u}\hat{v} & H + (1-\gamma)\hat{v}^2 & \gamma\hat{v} \end{pmatrix},$$

$$C(\hat{\boldsymbol{U}}) = \frac{1}{Re\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}\hat{u} & \frac{4}{3} & 0 & 0 \\ -\hat{v} & 0 & 1 & 0 \\ -\frac{4}{3}\hat{u}^2 - \hat{v}^2 + \frac{\gamma}{Pr}q - \frac{c^2}{(\gamma-1)Pr} & (\frac{4}{3} - \frac{\gamma}{Pr})\hat{u} & (1 - \frac{\gamma}{Pr})\hat{v} & \frac{\gamma}{Pr} \end{pmatrix},$$

$$D(\hat{\boldsymbol{U}}) = \frac{1}{Re\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\hat{u} & 1 & 0 & 0 \\ -\frac{4}{3}\hat{v} & 0 & \frac{4}{3} & 0 \\ -\hat{u}^2 - \frac{4}{3}\hat{v}^2 + \frac{\gamma}{Pr}q - \frac{c^2}{(\gamma-1)Pr} & (1 - \frac{\gamma}{Pr})\hat{u} & (\frac{4}{3} - \frac{\gamma}{Pr})\hat{v} & \frac{\gamma}{Pr} \end{pmatrix},$$

and

$$E(\hat{\boldsymbol{U}}) = \frac{1}{3Re\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\hat{v} & 0 & 1 & 0 \\ -\hat{u} & 1 & 0 & 0 \\ -2\hat{u}\hat{v} & \hat{v} & \hat{u} & 0 \end{pmatrix}.$$

The additional nonlinear terms are given as

$$
\begin{aligned}
NLT_1 =& 0, \\
NLT_2 =& \frac{1}{Re\rho^2}\left(2\hat{u}\rho_{\hat{y}}^2 - 2\rho_{\hat{y}}(\rho\hat{u})_{\hat{y}} + \frac{8}{3}\hat{u}\rho_{\hat{x}}^2 - \frac{8}{3}\rho_{\hat{x}}(\rho\hat{u})_{\hat{x}} + \frac{2}{3}\hat{v}\rho_{\hat{x}}\rho_{\hat{y}}\right. \\
& \left. -\frac{1}{3}\rho_{\hat{x}}(\rho\hat{v})_{\hat{y}} - \frac{1}{3}\rho_{\hat{y}}(\rho\hat{v})_{\hat{x}}\right), \\
NLT_3 =& \frac{1}{Re\rho^2}\left(2\hat{v}\rho_{\hat{x}}^2 - 2\rho_{\hat{x}}(\rho\hat{v})_{\hat{x}} + \frac{8}{3}\hat{v}\rho_{\hat{y}}^2 - \frac{8}{3}\rho_{\hat{y}}(\rho\hat{v})_{\hat{y}} + \frac{2}{3}\hat{u}\rho_{\hat{x}}\rho_{\hat{y}}\right. \\
& \left. -\frac{1}{3}\rho_{\hat{x}}(\rho\hat{u})_{\hat{y}} - \frac{1}{3}\rho_{\hat{y}}(\rho\hat{u})_{\hat{x}}\right), \\
NLT_4 =& \frac{1}{Re\rho^2}\left(\left(3\hat{u}^2 + 4\hat{v}^2 - 4\frac{\gamma}{Pr}q + \frac{2c^2}{(\gamma-1)Pr}\right)\rho_{\hat{y}}^2 + 4\hat{u}\left(\frac{\gamma}{Pr} - 1\right)\rho_{\hat{y}}(\rho\hat{u})_{\hat{y}}\right. \\
& + \left(1 - \frac{\gamma}{Pr}\right)(\rho\hat{u})_{\hat{y}}^2 + \left(\frac{4\gamma}{Pr} - \frac{16}{3}\right)\hat{v}\rho_{\hat{y}}(\rho\hat{v})_{\hat{y}} + \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)(\rho\hat{v})_{\hat{y}}^2 - \frac{2\gamma}{Pr}\rho_{\hat{y}}E_{\hat{y}} \\
& + \left(3\hat{v}^2 + 4\hat{u}^2 - 4\frac{\gamma}{Pr}q + \frac{2c^2}{(\gamma-1)Pr}\right)\rho_{\hat{x}}^2 + 4\hat{v}\left(\frac{\gamma}{Pr} - 1\right)\rho_{\hat{x}}(\rho\hat{v})_{\hat{x}} \\
& + \left(1 - \frac{\gamma}{Pr}\right)(\rho\hat{v})_{\hat{x}}^2 + \left(\frac{4\gamma}{Pr} - \frac{16}{3}\right)\hat{u}\rho_{\hat{x}}(\rho\hat{u})_{\hat{x}} + \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)(\rho\hat{u})_{\hat{x}}^2 - \frac{2\gamma}{Pr}\rho_{\hat{x}}E_{\hat{x}} \\
& + 2\hat{u}\hat{v}\rho_{\hat{x}}\rho_{\hat{y}} - \frac{7}{3}\hat{v}\rho_{\hat{x}}(\rho\hat{u})_{\hat{y}} + \hat{u}\rho_{\hat{x}}(\rho\hat{v})_{\hat{y}} + \hat{v}\rho_{\hat{y}}(\rho\hat{u})_{\hat{x}} - \frac{4}{3}(\rho\hat{u})_{\hat{x}}(\rho\hat{v})_{\hat{y}} \\
& \left. -\frac{7}{3}\hat{u}\rho_{\hat{y}}(\rho\hat{v})_{\hat{x}} + 2(\rho\hat{u})_{\hat{y}}(\rho\hat{v})_{\hat{x}}\right).
\end{aligned}
$$

We follow the idea of the one-dimensional case. Namely, we consider convection- and diffusion-dominated cases, and make a convex combination of them.

2.5.1 Convection-dominated case

For the convection-dominated case, we follow the idea in [31]. Assume the matrix $A(\hat{\boldsymbol{U}})$ has four eigenvalues $\lambda_1 = \hat{u} - c, \lambda_2 = \lambda_3 = \hat{u}, \lambda_4 = \hat{u} + c$, and a complete set of left eigenvectors $\{\boldsymbol{l}_1, \boldsymbol{l}_2, \boldsymbol{l}_3, \boldsymbol{l}_4\}$ which forms the matrix

$$
L = \begin{pmatrix} \boldsymbol{l}_1 \\ \boldsymbol{l}_2 \\ \boldsymbol{l}_3 \\ \boldsymbol{l}_4 \end{pmatrix} = \begin{pmatrix} l_{1,1} & l_{1,2} & l_{1,3} & l_{1,4} \\ l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\ l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\ l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4} \end{pmatrix}.
$$

We have $LAL^{-1} = \Lambda = diag(\lambda_1, \lambda_2, \lambda_3, \lambda_4)$ on the boundary. Let us assume $\lambda_1 < 0$, $\lambda_m \geq 0, \quad m = 2, 3, 4$. With the characteristic decomposition $\boldsymbol{V} = L\hat{\boldsymbol{U}}$, the equation

(2.43) at P_0 becomes

$$\boldsymbol{V}_t + \Lambda \boldsymbol{V}_{\hat{x}} = L\left(-B(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{y}} + C(\hat{\boldsymbol{U}})\hat{\boldsymbol{U}}_{\hat{x}\hat{x}} + D(\hat{\boldsymbol{U}})\boldsymbol{U}_{\hat{y}\hat{y}} + E(\hat{\boldsymbol{U}})\boldsymbol{U}_{\hat{x}\hat{y}} + \boldsymbol{NLT}\right) = \boldsymbol{RHS^V}. \tag{2.44}$$

As before, we take the given boundary condition for the convection-dominated case sequentially from the following ordered list: E, ρu, ρv and ρ, until the required number is reached. In this case, we should give one boundary condition and we take $\hat{U}_4 = g_4(t)$.

For the outgoing characteristic variables V_2, V_3, V_4, we use extrapolation to obtain $(V_m)^{ext}, (V_m)^{ext}_{\hat{x}}, (V_m)^{ext}_{\hat{x}\hat{x}}$, $m = 2, 3, 4$ on the boundary. Then we can obtain $\partial^{(0)}_{\hat{x}}\hat{\boldsymbol{U}}_c$ through the following equations

$$\begin{cases} \boldsymbol{l}_2 \cdot \partial^{(0)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_2)^{ext}, \\ \boldsymbol{l}_3 \cdot \partial^{(0)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_3)^{ext}, \\ \boldsymbol{l}_4 \cdot \partial^{(0)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_4)^{ext}, \\ \partial^{(0)}_{\hat{x}}(\hat{U}_c)_4 = g_4(t). \end{cases} \tag{2.45}$$

Next, we would like to find the first order derivatives $\partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_c$ by the inverse Lax-Wendroff method. Since the boundary conditions for $\hat{U}_4$ are prescribed, combining with the information of the outgoing characteristic variable V_2, V_3, V_4, we have

$$\begin{cases} \boldsymbol{l}_2 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_2)^{ext}_{\hat{x}}, \\ \boldsymbol{l}_3 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_3)^{ext}_{\hat{x}}, \\ \boldsymbol{l}_4 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_c = (V_4)^{ext}_{\hat{x}}, \\ A_4 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_c = -g_4'(t) + RHS^{\boldsymbol{V}}_4 \end{cases} \tag{2.46}$$

where A_m is the m-th row vector of the matrix A and $RHS^{\boldsymbol{V}}_m$ is the m-th entry of the vector $\boldsymbol{RHS^V}$ in (2.44), for $m = 1, 2, 3, 4$. The second order derivative $\partial^{(2)}_{\hat{x}}\hat{\boldsymbol{U}}_c$ is obtained by extrapolation.

2.5.2 Diffusion-dominated case

We consider this incompletely parabolic system as a scalar hyperbolic equation and a parabolic system. First, we diagonalize the matrix in front of the second order derivative terms on the boundary and obtain

$$\boldsymbol{W}_t + A'\boldsymbol{W}_{\hat{x}} + L'B\hat{\boldsymbol{U}}_{\hat{y}} = \Lambda'\boldsymbol{W}_{\hat{x}\hat{x}} + L'D\hat{\boldsymbol{U}}_{\hat{y}\hat{y}} + L'E\hat{\boldsymbol{U}}_{\hat{x}\hat{y}} + L'\boldsymbol{NLT}, \tag{2.47}$$

where

$$\boldsymbol{W} = L'\boldsymbol{U}, \quad L'BL'^{-1} = \Lambda' = \frac{1}{Re\rho} diag\left(0, 1, \frac{4}{3}, \frac{\gamma}{Pr}\right), A' = L'AL'^{-1},$$

and

$$A' = \begin{pmatrix} \hat{u} & 0 & -\frac{c}{\gamma} & 0 \\ 0 & \hat{u} & 0 & 0 \\ -c & 0 & \hat{u} & -c \\ 0 & 0 & -\frac{\gamma-1}{\gamma}c & \hat{u} \end{pmatrix}, \quad L' = \begin{pmatrix} \boldsymbol{l}'_1 \\ \boldsymbol{l}'_2 \\ \boldsymbol{l}'_3 \\ \boldsymbol{l}'_4 \end{pmatrix} = \begin{pmatrix} l'_{1,1} & l'_{1,2} & l'_{1,3} & l'_{1,4} \\ l'_{2,1} & l'_{2,2} & l'_{2,3} & l'_{2,4} \\ l'_{3,1} & l'_{3,2} & l'_{3,3} & l'_{3,4} \\ l'_{4,1} & l'_{4,2} & l'_{4,3} & l'_{4,4} \end{pmatrix},$$

where $\{\boldsymbol{l}'_1, \boldsymbol{l}'_2, \boldsymbol{l}'_3, \boldsymbol{l}'_4\}$ is a complete set of left eigenvectors of the matrix $C(\hat{\boldsymbol{U}})$ on the boundary. We rewrite the system (2.47) as

$$\begin{aligned} \boldsymbol{W}_t + A'\boldsymbol{W}_{\hat{x}} &= \Lambda'\boldsymbol{W}_{\hat{x}\hat{x}} + L'(-B\hat{\boldsymbol{U}}_{\hat{y}} + D\hat{\boldsymbol{U}}_{\hat{y}\hat{y}} + E\boldsymbol{U}_{\hat{x}\hat{y}} + \boldsymbol{NLT}) \\ &= \Lambda'\boldsymbol{W}_{\hat{x}\hat{x}} + \boldsymbol{RHS}^{\boldsymbol{W}}. \end{aligned} \tag{2.48}$$

We treat the incomplete parabolic equations (2.48) as a scalar hyperbolic equation and a parabolic system. Since the second order derivatives appear in the momentum and energy equations in (2.41), we take $\rho\hat{u}, \rho\hat{v}, E$ as the first choice of boundary conditions, and the boundary condition for ρ depends on the sign of the flow speed $\hat{u}$. If $\hat{u} < 0$, then we have four boundary conditions $\boldsymbol{g}(t)$. For simplicity we rewrite (2.43) as

$$\hat{\boldsymbol{U}}_t + A\hat{\boldsymbol{U}}_{\hat{x}} - C\hat{\boldsymbol{U}}_{\hat{x}\hat{x}} = -B\hat{\boldsymbol{U}}_{\hat{y}} + D\hat{\boldsymbol{U}}_{\hat{y}\hat{y}} + E\hat{\boldsymbol{U}}_{\hat{x}\hat{y}} + \boldsymbol{NLT} = \boldsymbol{RHS}^{\hat{\boldsymbol{U}}}$$

Our algorithm obtain the derivatives of each order is as follows.

For $\hat{u} < 0$, we have

$$\partial^{(0)}_{\hat{x}}\hat{\boldsymbol{U}}_d = \boldsymbol{g}(t), \tag{2.49}$$

$$\begin{cases} A_1 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_d = -g'_1(t) + C_1 \cdot \hat{\boldsymbol{U}}^{ext}_{\hat{x}\hat{x}} + RHS^{\hat{\boldsymbol{U}}}_1, \\ \boldsymbol{l}'_2 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_d = (W_2)^{ext}_{\hat{x}}, \\ \boldsymbol{l}'_3 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_d = (W_3)^{ext}_{\hat{x}}, \\ \boldsymbol{l}'_4 \cdot \partial^{(1)}_{\hat{x}}\hat{\boldsymbol{U}}_d = (W_4)^{ext}_{\hat{x}}, \end{cases} \tag{2.50}$$

$$\begin{cases} \boldsymbol{l}_1' \cdot \partial_{\hat{x}}^{(2)} \hat{\boldsymbol{U}}_d = (W_1)_{\hat{x}\hat{x}}^{ext}, \\ B_2 \cdot \partial_{\hat{x}}^{(2)} \hat{\boldsymbol{U}}_d = g_2'(t) + A_2 \cdot \hat{\boldsymbol{U}}_{\hat{x}}^{ext} + RHS_2^{\hat{\boldsymbol{U}}}, \\ B_3 \cdot \partial_{\hat{x}}^{(2)} \hat{\boldsymbol{U}}_d = g_3'(t) + A_3 \cdot \hat{\boldsymbol{U}}_{\hat{x}}^{ext} + RHS_3^{\hat{\boldsymbol{U}}}, \\ B_4 \cdot \partial_{\hat{x}}^{(2)} \hat{\boldsymbol{U}}_d = g_4'(t) + A_4 \cdot \hat{\boldsymbol{U}}_{\hat{x}}^{ext} + RHS_4^{\hat{\boldsymbol{U}}}, \end{cases} \tag{2.51}$$

For $\hat{u} \geq 0$, we only have three boundary conditions and the equations (2.49) and (2.50) are replaced by (2.52) and (2.53) to get $\partial_{\hat{x}}^{(0)} \hat{\boldsymbol{U}}_d$ and $\partial_{\hat{x}}^{(1)} \hat{\boldsymbol{U}}_d$, while (2.51) stays the same. That is

$$\begin{cases} \partial_{\hat{x}}^{(0)} (\hat{U}_d)_1 = \hat{U}_{\hat{x}}^{ext}, \\ \partial_{\hat{x}}^{(0)} (\hat{U}_d)_2 = g_2(t), \\ \partial_{\hat{x}}^{(0)} (\hat{U}_d)_3 = g_3(t), \\ \partial_{\hat{x}}^{(0)} (\hat{U}_d)_4 = g_4(t). \end{cases} \tag{2.52}$$

$$\partial_{\hat{x}}^{(1)} \hat{\boldsymbol{U}}_d = \hat{\boldsymbol{U}}_{\hat{x}}^{ext}. \tag{2.53}$$

2.5.3 General convection-diffusion case

Now we combine $\partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_c$ and $\partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_d$ together. Following the one-dimensional system case, we define

$$\begin{aligned} a_1 &= (A_{1,1}'^2 + A_{1,2}'^2 + A_{1,3}'^2 + A_{1,4}'^2)\Delta x^2, \quad \varepsilon_1 = 3(\lambda_2'^2 + \lambda_3'^2 + \lambda_4'^2), \quad \alpha_1 = \frac{a_1}{a_1 + \varepsilon_1}, \\ a_3 &= (A_{3,1}'^2 + A_{3,2}'^2 + A_{3,3}'^2 + A_{3,4}'^2)\Delta x^2, \quad \varepsilon_3 = 9\lambda_3'^2, \quad \alpha_3 = \frac{a_3}{a_3 + \varepsilon_3}, \\ a_4 &= (A_{4,1}'^2 + A_{4,2}'^2 + A_{4,3}'^2 + A_{4,4}'^2)\Delta x^2, \quad \varepsilon_4 = 9\lambda_4'^2, \quad \alpha_4 = \frac{a_4}{a_4 + \varepsilon_4}, \\ a_2 &= \tfrac{1}{3}(a_1 + a_3 + a_4), \quad \varepsilon_2 = 9\lambda_2'^2, \quad \alpha_2 = \frac{a_2}{a_2 + \varepsilon_2}, \end{aligned} \tag{2.54}$$

where $A_{i,j}'$ is the (i,j)-th entry of the matrix A' and $\{\lambda_m', m = 2,3,4\}$ are the eigenvalues of Λ' in (2.47). Here we again take ε_1 as the average of $\varepsilon_2, \varepsilon_3$ and ε_4, for the same reason as in the one-dimensional case. Also here we take a_2 as the average of a_1, a_3, a_4 for the same reason.

We make a transform of $\partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_c$ and $\partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_d$,

$$\partial_{\hat{x}}^{(k)} \boldsymbol{W}_c = L' \partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_c, \quad \partial_{\hat{x}}^{(k)} \boldsymbol{W}_d = L' \partial_{\hat{x}}^{(k)} \hat{\boldsymbol{U}}_d, k = 0,1,2, \tag{2.55}$$

and make a convex combination of them

$$\partial_{\hat{x}}^{(k)} W_i = \alpha_i \partial_{\hat{x}}^{(k)} (W_c)_i + (1 - \alpha_i) \partial_{\hat{x}}^{(k)} (W_d)_i, \quad i = 1,2,3,4, \quad k = 0,1,2. \tag{2.56}$$

After we get $\partial_{\hat{x}}^{(k)}\boldsymbol{W}, k = 0, 1, 2$, we take a transform back and obtain

$$\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}} = L'^{-1}\partial_{\hat{x}}^{(k)}\boldsymbol{W}, \quad k = 0, 1, 2. \tag{2.57}$$

Then we can use the Taylor expansion with these derivatives to define the values of the numerical solution at the ghost points.

Let us summarize our algorithm of the boundary treatment for two-dimensional Navier-Stokes systems.

Algorithm 3:

1. Set up a local coordinate (2.40) and introduce the new variables $(\rho, \rho\hat{u}, \rho\hat{v}, E)$, then we have (2.41). Consider (2.41) instead of (2.39).

2. For the convection-dominated case:

   - Compute the eigenvalues of A in (2.43) to decide the inflow and outflow boundary conditions. Use the characteristic decomposition on A and get the equation (2.44).

   - For any outgoing characteristic variable V_j, we use extrapolation to get the $\{(V_j)^{ext}, (V_j)_{\hat{x}}^{ext}, (V_j)_{\hat{x}\hat{x}}^{ext}\}$. With the prescribed boundary conditions for the incoming characteristic variables and extrapolated values $(V_j)^{ext}$, we can get $\partial_{\hat{x}}^{(0)}\hat{\boldsymbol{U}}_c$ by solving a linear system, such as (2.45).

   - With the inverse Lax-Wendroff procedure for the incoming characteristic variables and extrapolated values $(V_j)_{\hat{x}}^{ext}$, we can obtain $\partial_{\hat{x}}^{(1)}\hat{\boldsymbol{U}}_c$ by solving a linear system, such as (2.46).

   - The second order derivative is obtained by extrapolation, $\partial_{\hat{x}}^{(2)}\hat{\boldsymbol{U}}_c = \hat{\boldsymbol{U}}_{\hat{x}\hat{x}}^{ext}$.

3. For the diffusion-dominated case, we have two different cases depending on the sign of $\hat{u}$. Here we consider $\hat{u} < 0$.

- We have four boundary conditions for $\partial_{\hat{x}}^{(0)}\hat{\boldsymbol{U}}_d$, as in (2.49).
- The first order derivative $\partial_{\hat{x}}^{(1)}\boldsymbol{U}_d$ are obtained by applying the inverse Lax-Wendroff method to the mass equation, together with the extrapolation equations for $\partial_{\hat{x}}^{(1)}(W_d)_m, m = 2, 3, 4$, as in (2.50).
- The second order derivative $\partial_{\hat{x}}^{(2)}\boldsymbol{W}_d$ are obtained by applying the inverse Lax-Wendroff method to momentum equation and energy equation, together with the extrapolation equation for $\partial_{\hat{x}}^{(2)}(W_d)_1$, as in (2.51).

4. To combine $\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}}_c$ and $\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}}_d$, $k = 0, 1, 2$:

   - Choose the coefficient $\{\alpha_j, j = 1, 2, 3, 4\}$ defined in (2.54).
   - Convert $\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}}_c$ and $\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}}_d$ into $\partial_{\hat{x}}^{(k)}\boldsymbol{W}_c$ and $\partial_{\hat{x}}^{(k)}\boldsymbol{W}_d$, $k = 0, 1, 2$, respectively, as in (2.55).
   - Then make a convex combination of the derivatives $\partial_{\hat{x}}^{(k)}\boldsymbol{W}_c$ and $\partial_{\hat{x}}^{(k)}\boldsymbol{W}_d$ to get the derivatives $\partial_{\hat{x}}^{(k)}\boldsymbol{W}$, $k = 0, 1, 2$, as in (2.56).
   - Then we transform from $\partial_{\hat{x}}^{(k)}\boldsymbol{W}$ back to $\partial_{\hat{x}}^{(k)}\hat{\boldsymbol{U}}$, $k = 0, 1, 2$, as in (2.57).

5. We assign the values of the ghost points by the Taylor expansion.

Remark: If $\hat{u} \geq 0$, the equations (2.49) and (2.50) in Step 3 need to be replaced by (2.52) and (2.53) to obtain $\partial_{\hat{x}}^{(0)}\hat{\boldsymbol{U}}_d$ and $\partial_{\hat{x}}^{(1)}\hat{\boldsymbol{U}}_d$, while other steps are unchanged.

3 Numerical examples

In this section we show numerical examples to demonstrate the effectiveness of our algorithm. Throughout this section, we use second order Taylor expansion to achieve third order accuracy of the boundary treatments.

3.1 One-dimensional examples

In this subsection we take $\delta_1 = 10^{-6}$ and $\delta_2 = 10^{-6}$ in (2.4) so that the physical boundaries do not coincide with the closest grid points but are very close to them. This is a typical "cut-cell" situation which is the most demanding for stability under normal CFL conditions.

Example 1. We start with the linear scalar convection-diffusion equation

$$\begin{cases} u_t + au_x = \varepsilon u_{xx}, \quad -1 < x < 1, \quad t > 0, \\ u(-1,t) = e^{-\varepsilon t}\sin(-1-at), \quad u(1,t) = e^{-\varepsilon t}\sin(1-at), \quad t > 0, \\ u(x,0) = \sin x, \quad -1 \le x \le 1, \end{cases} \tag{3.1}$$

with the exact solution given as

$$u(x,t) = e^{-\varepsilon t}\sin(x-at).$$

The final time is $t = 1$ and the time step is taken as $\Delta t = 0.6\Delta x^2/(a\Delta x + 2\varepsilon)$. We consider several pairs of (a, ε).

Our algorithm runs in a stable fashion for all cases, indicating that the "cut-cell" difficulty does not arise. In Tables 1–3, we can clearly see third order accuracy, which demonstrates that our method performs well in both convection-dominated and diffusion-dominated situations.

Table 1: Linear equation (3.1) with $a = 1$ and $\varepsilon = 10^{-6}$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	4.71E-04	–	5.23E-04	–	8.18E-04	–
20	5.58E-05	3.08	6.64E-05	2.98	1.14E-04	2.84
40	6.94E-06	3.01	8.31E-06	3.00	1.54E-05	2.89
80	8.63E-07	3.01	1.03E-06	3.01	2.03E-06	2.93
160	1.07E-07	3.01	1.29E-07	3.01	2.62E-07	2.95
320	1.34E-08	3.00	1.60E-08	3.00	3.33E-08	2.97
640	1.67E-09	3.00	2.00E-09	3.00	4.21E-09	2.98

Table 2: Linear equation (3.1) with $a = 0.1$ and $\varepsilon = 0.1$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	2.93E-05	–	5.04E-05	–	1.49E-04	–
20	7.34E-06	2.00	1.15E-05	2.13	3.48E-05	2.10
40	1.25E-06	2.55	1.83E-06	2.66	5.41E-06	2.69
80	1.80E-07	2.80	2.54E-07	2.85	7.45E-07	2.86
160	2.39E-08	2.91	3.34E-08	2.93	9.75E-08	2.93
320	3.08E-09	2.96	4.28E-09	2.96	1.25E-08	2.97
640	3.90E-10	2.98	5.41E-10	2.98	1.58E-09	2.98

Table 3: Linear equation (3.1) with $a = 10^{-6}$ and $\varepsilon = 1$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	2.84E-05	–	3.29E-05	–	5.34E-05	–
20	6.18E-06	2.20	7.07E-06	2.22	1.18E-05	2.18
40	9.65E-07	2.68	1.10E-06	2.69	1.87E-06	2.66
80	1.33E-07	2.86	1.52E-07	2.86	2.60E-07	2.84
160	1.75E-08	2.93	1.99E-08	2.93	3.42E-08	2.93
320	2.23E-09	2.97	2.54E-09	2.97	4.39E-09	2.96
640	2.82E-10	2.98	3.21E-10	2.98	5.56E-10	2.98

Example 2. The viscous Burgers equation is given by

$$\begin{cases} u_t + \frac{1}{2}(u^2)_x = \varepsilon u_{xx}, \quad -1 < x < 1, \quad t > 0, \\ u(-1,t) = g_1(t), \quad u(1,t) = g_2(t), \quad t > 0, \\ u(x,0) = u_0(x), \quad -1 \le x \le 1, \end{cases} \tag{3.2}$$

where

$$g_1(t) = 0.5 - 0.5\tanh\left(-1.5 - 0.5t)/(4\varepsilon)\right),$$

$$g_2(t) = 0.5 - 0.5\tanh\left((0.5 - 0.5t)/(4\varepsilon)\right),$$

$$u_0(x) = 0.5 - 0.5\tanh\left((x - 0.5)/(4\varepsilon)\right).$$

The exact solution is given as

$$u(x,t) = 0.5 - 0.5\tanh\left(\frac{x - 0.5 - 0.5t}{4\varepsilon}\right).$$

In Table 4, we can see that, for a small $\varepsilon = 0.01$, the scheme achieves third order accuracy eventually with mesh refinements. This is reasonable because when ε is small, there is a sharp layer near the right boundary $x = 1$ at time $t = 1$. We need fine meshes to resolve the sharp layer. As ε becomes larger, the solution becomes smoother and the convergence achieves third order on coarser grids. This is exactly what we observe in Table 5.

Table 4: Viscous Burgers equation (3.2) with $\varepsilon = 0.01$. The sharp layer near the right boundary requires fine meshes to resolve.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	9.74E-01	–	1.36E+00	–	2.81E+00	–
20	1.02E-02	6.58	2.64E-02	5.69	9.64E-02	4.86
40	3.82E-03	1.41	1.36E-02	0.95	7.11E-02	0.44
80	7.92E-04	2.27	3.98E-03	1.78	2.98E-02	1.26
160	1.11E-04	2.83	5.91E-04	2.75	4.85E-03	2.62
320	1.28E-05	3.11	6.68E-05	3.15	5.39E-04	3.17
640	1.58E-06	3.02	8.28E-06	3.01	7.00E-05	2.94

Table 5: Viscous Burgers equation (3.2) with $\varepsilon = 1$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	4.81E-06	–	5.91E-06	–	1.15E-05	–
20	5.21E-07	3.21	6.66E-07	3.15	1.43E-06	3.02
40	6.21E-08	3.07	7.96E-08	3.06	1.79E-07	2.99
80	7.63E-09	3.02	9.76E-09	3.03	2.25E-08	2.99
160	9.48E-10	3.01	1.21E-09	3.01	2.82E-09	3.00
320	1.18E-10	3.00	1.51E-10	3.01	3.53E-10	3.00
640	1.49E-11	2.99	1.89E-11	2.99	4.42E-11	3.00

Example 3. We consider a linear system (2.12) with $d = 2$. We can always diagonalize either of the matrices A and B. For simplicity we take B as a diagonal matrix, then we

have a linear system

$$\begin{pmatrix} u \\ v \end{pmatrix}_t + \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_x = \begin{pmatrix} \varepsilon_1 & 0 \\ 0 & \varepsilon_2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}_{xx}, \tag{3.3}$$

where $\{a_{i,j}, i = 1, 2, j = 1, 2\}$ and $\varepsilon_1, \varepsilon_2$ are constants. We add suitable source terms to (3.3) so that it has a smooth solution given by

$$u(x,t) = e^{-\varepsilon_1 t} \sin(x - a_{1,1}t), \quad v(x,t) = e^{-\varepsilon_2 t} \cos(x - a_{2,2}t).$$

We test four combinations of (A_i, B_j), $i = 1, 2$ and $j = 1, 2$.

$$A_1 = \begin{pmatrix} 3 & 0.5 \\ 0.5 & 2 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 3 & 0.5 \\ 0.5 & -2 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 10^{-5} & 0 \\ 0 & 10^{-6} \end{pmatrix}, \quad B_2 = \begin{pmatrix} 0.8 & 0 \\ 0 & 1 \end{pmatrix}.$$

Here A_1 has two positive eigenvalues, while A_2 has one positive and one negative eigenvalue. B_1 has two small eigenvalues, leading to convection-dominated equations. B_2 has two normal-sized eigenvalues, leading to diffusion-dominated equations. The final time we take is $t = 1$, and the time step is

$$\Delta t = 0.4\Delta x^2 / (\sigma(A)\Delta x + 2\max(\varepsilon_1, \varepsilon_2)),$$

where $\sigma(A)$ is the spectral radius of A (the largest eigenvalue in magnitude of A).

To save space, we show the errors and orders of convergence for u only. From Tables 6–9, we can observe the designed third order accuracy for all cases.

Table 6: Linear system (3.3) with (A_1, B_1). A_1 has two positive eigenvalues and B_1 has two small eigenvalues.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	5.38E-04	–	6.15E-04	–	1.22E-03	–
20	7.01E-05	2.94	7.89E-05	2.96	1.58E-04	2.96
40	8.81E-06	2.99	9.85E-06	3.00	2.02E-05	2.97
80	1.10E-06	3.01	1.22E-06	3.01	2.51E-06	3.01
160	1.37E-07	3.01	1.52E-07	3.01	3.10E-07	3.01
320	1.70E-08	3.00	1.89E-08	3.01	3.88E-08	3.00
640	2.12E-09	3.00	2.36E-09	3.00	4.84E-09	3.00

Table 7: Linear system (3.3) with (A_1, B_2). A_1 has two positive eigenvalues and B_2 has two normal-sized eigenvalues.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	2.57E-04	–	2.82E-04	–	4.18E-04	–
20	3.51E-05	2.88	3.80E-05	2.89	5.41E-05	2.95
40	4.83E-06	2.86	5.14E-06	2.89	7.04E-06	2.94
80	6.38E-07	2.92	6.75E-07	2.93	9.04E-07	2.96
160	8.22E-08	2.96	8.67E-08	2.96	1.15E-07	2.98
320	1.04E-08	2.98	1.10E-08	2.98	1.45E-08	2.99
640	1.32E-09	2.99	1.39E-09	2.99	1.82E-09	2.99

Table 8: Linear system (3.3) with (A_2, B_1). A_2 has one positive and one negative eigenvalue; B_1 has two small eigenvalues.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	4.49E-04	–	4.93E-04	–	6.76E-04	–
20	5.96E-05	2.91	6.55E-05	2.91	9.10E-05	2.89
40	7.48E-06	2.99	8.22E-06	3.00	1.14E-05	3.00
80	9.30E-07	3.01	1.02E-06	3.01	1.41E-06	3.01
160	1.16E-07	3.01	1.27E-07	3.01	1.75E-07	3.01
320	1.44E-08	3.00	1.58E-08	3.01	2.18E-08	3.01
640	1.80E-09	3.00	1.96E-09	3.00	2.71E-09	3.00

Table 9: Linear system (3.3) with (A_2, B_2). A_2 has one positive and one negative eigenvalue; B_2 has two normal-sized eigenvalues.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	2.33E-04	–	2.55E-04	–	3.87E-04	–
20	3.36E-05	2.79	3.63E-05	2.81	5.14E-05	2.91
40	4.85E-06	2.79	5.14E-06	2.82	6.97E-06	2.88
80	6.57E-07	2.88	6.93E-07	2.89	9.15E-07	2.93
160	8.58E-08	2.94	9.01E-08	2.94	1.17E-07	2.96
320	1.10E-08	2.97	1.15E-08	2.97	1.49E-08	2.98
640	1.39E-09	2.97	1.46E-09	2.98	1.89E-09	2.98

Example 4. We consider the one-dimensional Navier-Stokes equations. We modify the system (2.24) with additional source terms so that we have an explicit exact solution to test accuracy. The modified system is

$$\begin{cases} \rho_t + (\rho u)_x = f_1(x,t) \\ (\rho u)_t + (\rho u^2 + p)_x = \frac{1}{Re}\left(\frac{4}{3}u\right)_{xx} + f_2(x,t) \\ E_t + (u(E+p))_x = \frac{1}{Re}\left[\frac{2}{3}(u^2)_{xx} + \frac{(c^2)_{xx}}{(\gamma-1)Pr}\right] + f_3(x,t) \end{cases} \tag{3.4}$$

with the exact solution

$$\rho(x,t) = \frac{e^{\sin t}}{1+3x^2}, \quad u(x,t) = 1 + 3x^2, \quad p(x,t) = pre.$$

Here pre is a constant to be determined which is used to adjust the number of the positive eigenvalues $\{u-c, u, u+c\}$. If we take $pre = 10$, then $\lambda_1 < 0, \lambda_2 > 0, \lambda_3 > 0$ at both the left and the right boundaries. Then we have two boundary conditions at the left boundary and one boundary condition at the right boundary in the convection-dominated case. Here we prescribe boundary conditions on $\rho u, E$ at $x = -1$ and on E at $x = 1$. In the diffusion-dominated case we have three boundary conditions at the left boundary and two boundary conditions at the right boundary. We prescribe boundary conditions on $\rho, \rho u, E$ at $x = -1$ and on $\rho u, E$ at $x = 1$. If we take $pre = 0.5$, then $\lambda_1 > 0, \lambda_2 > 0, \lambda_3 > 0$ on both the left and the right boundaries. Then we have three boundary conditions at the left boundary and no boundary condition at the right boundary in the convection-dominated case, and three boundary conditions at the left boundary and two boundary conditions at the right boundary for the diffusion-dominated case. We document our choice of boundary conditions in Table 10. The final time is $t = 2$, and the time step is chosen as

$$\Delta t = 0.6\Delta x^2 / (\max\{|\lambda_1|, |\lambda_2|, |\lambda_3|\}\Delta x + 2\max\{\Lambda'_{22}, \Lambda'_{33}\}).$$

We take several different values of Reynolds number Re to demonstrate the effectiveness of our algorithm. The error tables we present here are for the density. In Tables 11–16, we can see third order accuracy for all the choices of Reynolds numbers $Re = 1$, 30, 10^6 in L^1 and L^2 errors and in most cases also in L^∞ errors.

Table 10: Choice of boundary conditions for the modified Navier-Stokes equations (3.4).

	$pre = 0.5$		$pre = 10$	
	$x = -1$	$x = 1$	$x = -1$	$x = 1$
convection-dominated	$\rho, \rho u, E$	no B.C.s	$\rho u, E$	E
diffusion-dominated	$\rho, \rho u, E$	$\rho u, E$	$\rho, \rho u, E$	$\rho u, E$

Table 11: Modified Navier-Stokes equations (3.4) with $pre = 0.5$ and $Re = 1$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	5.55E-02	–	6.72E-02	–	1.35E-01	–
20	5.28E-03	3.39	7.53E-03	3.16	1.70E-02	2.99
40	6.14E-04	3.11	8.79E-04	3.10	1.98E-03	3.10
80	7.27E-05	3.08	1.03E-04	3.09	2.26E-04	3.13
160	8.82E-06	3.04	1.25E-05	3.05	2.65E-05	3.09
320	1.08E-06	3.02	1.53E-06	3.02	3.27E-06	3.02
640	1.35E-07	3.01	1.90E-07	3.01	4.08E-07	3.00

Table 12: Modified Navier-Stokes equations (3.4) with $pre = 0.5$ and $Re = 30$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	1.52E-01	–	1.83E-01	–	3.35E-01	–
20	1.63E-02	3.22	2.31E-02	2.98	4.65E-02	2.85
40	2.00E-03	3.03	2.83E-03	3.03	5.94E-03	2.97
80	2.39E-04	3.06	3.34E-04	3.08	7.03E-04	3.08
160	2.94E-05	3.03	4.05E-05	3.04	8.34E-05	3.08
320	3.93E-06	2.90	5.62E-06	2.85	3.02E-05	1.46
640	5.11E-07	2.95	7.72E-07	2.86	5.66E-06	2.42

Table 13: Modified Navier-Stokes equations (3.4) with $pre = 0.5$ and $Re = 10^6$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	3.50E-01	–	3.93E-01	–	6.98E-01	–
20	1.75E-02	4.33	2.39E-02	4.04	4.72E-02	3.89
40	1.96E-03	3.16	2.72E-03	3.13	5.73E-03	3.04
80	2.23E-04	3.13	3.19E-04	3.09	7.14E-04	3.00
160	2.68E-05	3.06	3.85E-05	3.05	8.55E-05	3.06
320	3.28E-06	3.03	4.71E-06	3.03	1.05E-05	3.03
640	4.06E-07	3.01	5.83E-07	3.01	1.29E-06	3.02

Table 14: Modified Navier-Stokes equations (3.4) with $pre = 10$ and $Re = 1$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	1.07E-01	–	1.35E-01	–	2.48E-01	–
20	5.76E-03	4.22	7.74E-03	4.13	1.94E-02	3.68
40	6.87E-04	3.07	9.11E-04	3.09	2.40E-03	3.02
80	8.25E-05	3.06	1.07E-04	3.09	2.75E-04	3.12
160	1.00E-05	3.04	1.28E-05	3.06	3.25E-05	3.08
320	1.24E-06	3.02	1.57E-06	3.03	3.94E-06	3.04
640	1.53E-07	3.01	1.94E-07	3.02	4.86E-07	3.02

Table 15: Modified Navier-Stokes equations (3.4) with $pre = 10$ and $Re = 30$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
10	5.25E-02	–	6.83E-01	–	1.50E-01	–
20	1.15E-02	2.19	1.31E-02	2.38	2.41E-02	2.64
40	6.75E-04	4.09	9.87E-04	3.73	2.43E-03	3.31
80	7.66E-05	3.14	1.18E-04	3.06	3.01E-04	3.02
160	9.17E-06	3.06	1.41E-05	3.06	3.81E-05	2.98
320	1.09E-06	3.07	1.67E-06	3.08	4.48E-06	3.09
640	1.35E-07	3.01	2.08E-07	3.01	5.58E-07	3.01

Table 16: Modified Navier-Stokes equations (3.4) with $pre = 10$ and $Re = 10^6$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
20	5.43E-03	–	6.85E-03	–	1.53E-02	–
40	7.12E-04	2.93	8.94E-04	2.94	1.84E-03	3.06
80	8.68E-05	3.04	1.11E-04	3.01	2.33E-04	2.98
160	1.06E-05	3.03	1.36E-05	3.02	2.94E-05	2.99
320	1.32E-06	3.00	1.70E-06	3.00	3.72E-06	2.98
640	1.65E-07	3.00	2.12E-07	3.00	4.63E-07	3.01

Example 5. We consider an additional example for the one-dimensional Navier-Stokes equations (3.4) when both convection-dominated region and diffusion-dominated region co-exist in the domain at the same time. We add suitable source terms so that the exact solution is

$$\rho(x,t) = 1 + 0.1\sin(x+t), \quad u(x,t) = 0.5 + 3(x+1)^2, \quad p(x,t) = \frac{u(x,t)^2}{4\gamma}.$$

From this exact solution we can verify that $\{u-c, u, u+c\}$ are positive on both the left boundary and the right boundary, and the flow speed $u(-1,t) = 0.5$ and $u(1,t) = 12.5$. We take Reynolds number $Re = 300$ and the CFL condition is the same as in Example 4. From (2.35) in Algorithm 2 for the one-dimensional Navier-Stokes equations, we can obtain the coefficients for combining convection-dominated and diffusion-dominated cases. When $N = 320$, it is diffusion-dominated on the left boundary and convection-dominated on the right boundary. We present the coefficients in Table 17 when $t = 0$.

Table 17: Coefficients obtained by (2.35), $Re = 300$, $N = 320$, $t = 0$.

	$x = -1$	$x = 1$
α_1	3.13E-02	0.97
α_2	6.65E-02	0.98
α_3	2.05E-02	0.95

Table 18: Modified Navier-Stokes equations (3.4) with $Re = 300$.

N	L^1 error	order	L^2 error	order	L^∞ error	order
20	2.98E-03	–	3.76E-03	–	8.31E-03	–
40	3.54E-04	3.08	3.76E-04	3.32	6.27E-04	3.73
80	4.58E-05	2.95	5.16E-05	2.87	7.00E-05	3.16
160	7.00E-06	2.71	7.33E-06	2.82	9.35E-06	2.90
320	8.99E-07	2.96	9.31E-07	2.98	1.14E-06	3.04
640	1.05E-07	3.10	1.10E-07	3.08	1.38E-07	3.05

From Table 18, we can see that our boundary treatment is stable and accurate when both convection-dominated region and diffusion-dominated region co-exist in the domain at the same time.

3.2 Two-dimensional examples

Example 6. We consider the two-dimensional Burgers equation

$$u_t + \frac{1}{2}(u^2)_x + \frac{1}{2}(u^2)_y = \varepsilon(u_{xx} + u_{yy}), \quad (x, y) \in [-1, 1] \times [-1, 1], \quad t > 0. \tag{3.5}$$

We add a source term to the equation so that we can have an explicit solution to the equation. The exact solution is

$$u(x, y, t) = e^{-\varepsilon t}(1 - 0.2 \sin x)(1 - 0.5 \cos y)$$

with the boundary conditions

$$g_1(y, t) = u(-1, y, t), \quad g_2(y, t) = u(1, y, t), \quad g_3(x, t) = u(x, -1, t), \quad g_4(x, t) = u(x, 1, t).$$

Our mesh is (x_i, y_j), $i = -2, -1, \cdots, N_x + 1, N_x + 2$, $j = -2, -1, \cdots, N_y + 1, N_y + 2$, where

$$-1 + \delta_1 \Delta x = x_0 < x_1 < \cdots < x_{N_x - 1} < x_{N_x} = 1 - \delta_2 \Delta x, \quad \delta_1 = \delta_2 = 10^{-6}$$

$$-1 + \delta_3 \Delta y = y_0 < y_1 < \cdots < y_{N_y - 1} < y_{N_y} = 1 - \delta_4 \Delta y, \quad \delta_3 = \delta_4 = 10^{-6}$$

The final time is $t = 1$. We take the time step as

$$\Delta t = 0.8/(2\varepsilon(\Delta x^{-2} + \Delta y^{-2}) + \max_{x\in[-1,1]} |u_0(x)|(\Delta x^{-1} + \Delta y^{-1})).$$

From the Tables 19–20 we can see the designed third order accuracy. Again the difficulty with the "cut-cell" problem does not arise.

Table 19: 2D Burgers equation (3.5) with $\varepsilon = 10^{-6}$.

$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	1.44E-04	–	1.79E-04	–	5.89E-04	–
20×20	2.02E-05	2.83	2.48E-05	2.85	8.27E-05	2.83
40×40	2.63E-06	2.94	3.23E-06	2.94	1.02E-05	3.01
80×80	3.35E-07	2.97	4.13E-07	2.97	1.28E-06	3.00
160×160	4.22E-08	2.99	5.22E-08	2.98	1.60E-07	3.00
320×320	5.30E-09	2.99	6.56E-09	2.99	1.99E-08	3.00

Table 20: 2D Burgers equation (3.5) with $\varepsilon = 1$.

$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	9.41E-05	–	1.05E-04	–	1.82E-04	–
20×20	1.06E-05	3.15	1.15E-05	3.19	1.92E-05	3.24
40×40	1.28E-06	3.05	1.37E-06	3.07	2.24E-06	3.10
80×80	1.58E-07	3.02	1.68E-07	3.03	2.72E-07	3.05
160×160	1.96E-08	3.01	2.09E-08	3.01	3.34E-08	3.02
320×320	2.45E-09	3.00	2.60E-09	3.00	4.15E-09	3.01

Example 7. We consider the two-dimensional Navier-Stokes equations (2.39). By adding additional source terms to the system, we have an explicit exact solution to test accuracy. The exact solution is given by

$$\rho(x, y, t) = e^{t \sin x \sin y}, \quad u(x, y, t) = 2 + 0.02(x^2 - \pi^2),$$
$$v(x, y, t) = 1 + 0.01(y^2 - \pi^2), \quad p(x, y, t) = 5.$$

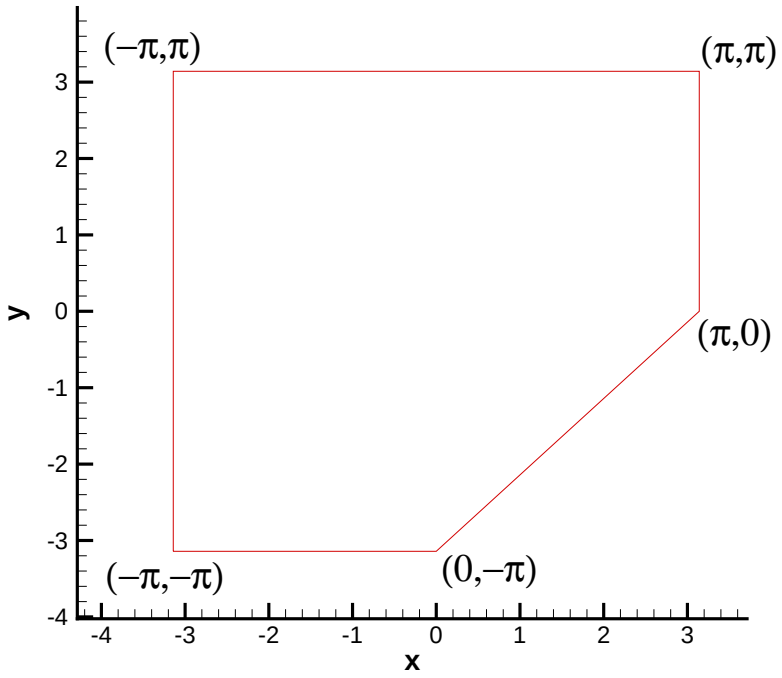


Figure 1: The computational domain of the two-dimensional Navier-Stokes equations (2.39).

The computational domain is a polygon shown in Figure 1. One edge of the polygon is inclined at 45 degree angle to the grid lines.

We take the mesh in the same way as in Example 6, the final time $t = 1$ and the time step

$$\Delta t = 0.6/\left(\max\{|\lambda_u|\}\frac{1}{\Delta x} + \max\{|\lambda_v|\}\frac{1}{\Delta y} + 6\max\{\lambda_{\Lambda'}\}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)\right), \qquad (3.6)$$

where $\lambda_u = \{u - c, u, u + c\}$, $\lambda_v = \{v - c, v, v + c\}$ and $\lambda_{\Lambda'} = \left\{\frac{1}{Re\rho}, \frac{4}{3Re\rho}, \frac{\gamma}{PrRe\rho}\right\}$.

In this example, along the line segment with two endpoints $(0, -\pi)$ and $(\pi, 0)$, we have $\hat{u} > 0$. Thus we have only one boundary condition in the convection-dominated case and three boundary conditions for the diffusion-dominated case. From Tables 21–23, we can again see third order accuracy for all the choices of Reynolds numbers $Re = 1$, 30, 10^6 in L^1 and L^2 errors and in most cases also in L^∞ errors.

Table 21: 2D Navier-Stokes equations (2.39) with $Re = 1$.

$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	3.08E-02	–	4.91E-02	–	1.90E-01	–
20×20	5.54E-03	2.48	9.15E-03	2.42	5.91E-02	1.69
40×40	4.20E-04	3.72	6.42E-04	3.83	3.02E-03	4.29
80×80	4.50E-05	3.22	7.14E-05	3.17	4.53E-04	2.74
160×160	5.17E-06	3.12	8.26E-06	3.11	5.32E-05	3.09
320×320	3.05E-07	3.05	1.00E-06	3.04	4.95E-06	3.43

Table 22: 2D Navier-Stokes equations (2.39) with $Re = 30$.

$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	1.56E-02	–	2.65E-02	–	1.16E-01	–
20×20	2.53E-03	2.63	4.20E-03	2.66	1.91E-02	2.60
40×40	3.40E-04	2.89	5.74E-04	2.87	2.66E-03	2.84
80×80	4.77E-05	2.83	8.77E-05	2.71	8.21E-04	1.70
160×160	6.22E-06	2.94	1.12E-05	2.97	1.13E-04	2.86
320×320	7.10E-07	3.13	1.16E-06	3.27	1.17E-05	3.27

Table 23: 2D Navier-Stokes equations (2.39) with $Re = 10^6$.

$N_x \times N_y$	L^1 error	order	L^2 error	order	L^∞ error	order
10×10	1.59E-02	–	2.66E-02	–	1.15E-01	–
20×20	2.76E-03	2.52	4.35E-03	2.61	1.93E-02	2.58
40×40	3.53E-04	2.97	5.65E-04	2.94	2.68E-03	2.84
80×80	4.38E-05	3.01	7.10E-05	2.99	3.40E-04	2.98
160×160	5.45E-06	3.01	8.87E-06	3.00	4.26E-05	3.00
320×320	6.79E-07	3.00	1.11E-06	3.00	5.31E-06	3.00

Example 8. Our last example is the flow past a circular cylinder positioned at the origin with radius 1. A Mach 3 flow is moving towards the cylinder from the left. One approach to impose the wall boundary condition is generating a boundary fitted mesh

[12]. To avoid generating such a boundary fitted mesh, we use our method on a Cartesian mesh.

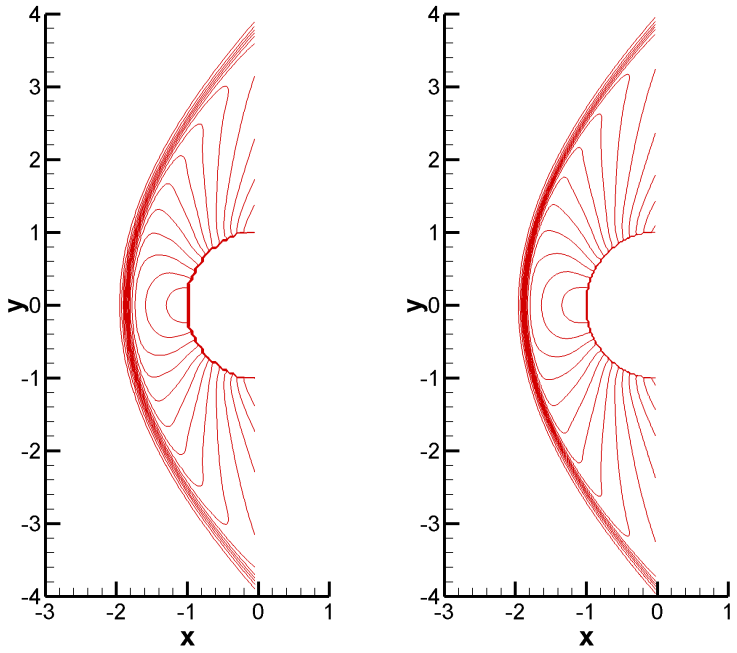
Since there is a shock (sharp layer) near the surface of the cylinder during the time evolution, we use a third order upwind-biased WENO approximation in the interior scheme and a WENO type extrapolation in the boundary condition. We take the domain as large as $[-3,3]\times[-6,6]$ to avoid possible issues associated with the far-field boundary conditions. Due to the symmetry of this problem, we only compute the upper half of the domain and use the symmetry condition at $y=0$. The time step is computed according to (3.6). We take the final time $t=40$ when the flow reaches a steady state in the region $[-3,0]\times[-4,4]$ which is of interest to us.

To make the initial condition compatible with the boundary condition, we take the initial condition as

$$\rho(x,y,0)=1.4,\quad u(x,y,0)=\begin{cases} x^2+y^2-1 & , \quad 1<x^2+y^2\le 4 \\ 3 & , \quad \text{elsewhere} \end{cases},$$
$$v(x,y,0)=0,\quad p(x,y,0)=1$$

The boundary condition for $x=-3$ is the supersonic inflow boundary condition. For the boundaries at $x=3$ and $y=6$ we use the free-stream data $(\rho,u,v,p)=(1.4,3,0,1)$. On the surface of the cylinder, we have a slip wall condition $u=0$ for the convection-dominated case. For the diffusion-dominated case, we prescribe the no-slip wall condition $u=v=0$ and isothermal wall condition $T_w=T_0$, where $T_0=2$.

The pressure contours at steady state are shown in Figures 2 and 3 for $Re=10$ and 10^8, respectively. In each figure, we display the results with the mesh sizes $\Delta x=\Delta y=\frac{1}{20}$ and $\frac{1}{40}$. Our algorithm runs in a stable fashion and the bow shock is well captured by our method. We display the pressure profile along the center line $y=0$ in Figure 4. The shock is smeared for the low Reynolds number case.

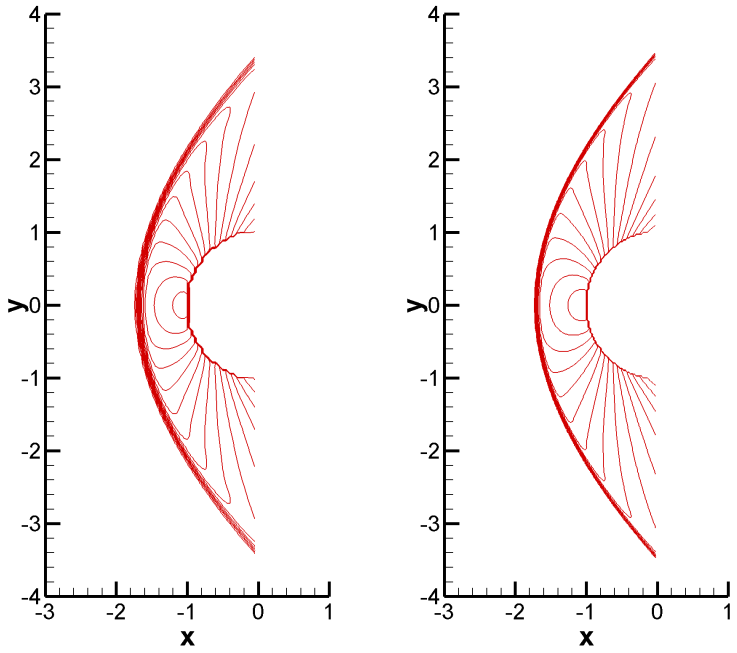


(a) $\Delta x = \Delta y = \frac{1}{20}$ (b) $\Delta x = \Delta y = \frac{1}{40}$

Figure 2: $Re = 10$, pressure contour of flow past a cylinder, 20 contours from 2 to 15

4 Concluding remarks

In this paper we consider stable and accurate numerical boundary conditions for high order finite difference schemes on Cartesian meshes to solve convection-diffusion equations in arbitrarily shaped domains. We extend the so-called inverse Lax-Wendroff procedure, developed in [28] for hyperbolic equations, to handle the convection-diffusion equations. Since totally different boundary treatments are needed for the diffusion-dominated and the convection-dominated cases, we have developed a careful combination of these two boundary treatments to obtain a stable and accurate boundary condition for general convection-diffusion equations. Our methodology can also handle incompletely parabolic systems, particularly the compressible Navier-Stokes equations, in a stable and accurate fashion. We provide extensive numerical tests for one- and two-dimensional problems, including the compressible Navier-Stokes equations on complex domains, to demonstrate



(a) $\Delta x = \Delta y = \frac{1}{20}$ (b) $\Delta x = \Delta y = \frac{1}{40}$

Figure 3: $Re = 10^8$, pressure contour of flow past a cylinder, 20 contours from 2 to 15

the good performance of our numerical boundary conditions.

For convenience we have only demonstrated and tested our methodology for the third order accurate schemes. However, we expect the methodology to work well for higher order schemes, which will be tested in future work. We have considered exclusively Dirichlet boundary conditions in this paper. Extensions to Neumann or Robin type boundary conditions, such as the adiabatic boundary condition for compressible Navier-Stokes equations, will also be investigated in future work.

References

[1] M.J. Berger, C. Helzel and R.J. Leveque, *h-box methods for the approximation of hyperbolic conservation laws on irregular grids*, SIAM Journal on Numerical Analysis, 41 (2003), 893-918.

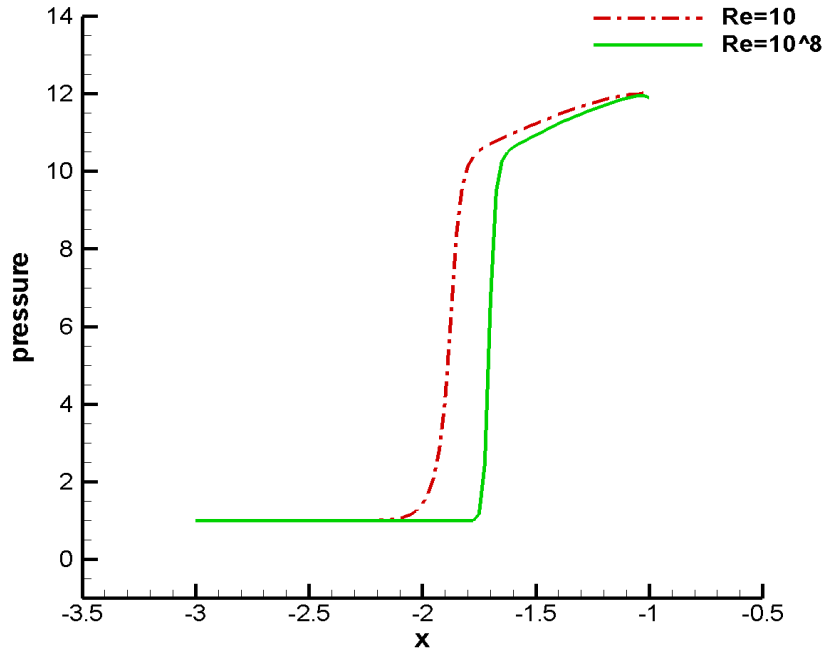


Figure 4: Pressure profiles along the center line $y = 0$, $\Delta x = \Delta y = \frac{1}{40}$. Solid line: $Re = 10^8$; Dashed line: $Re = 10$. The shock is smeared when the Reynolds number is low.

[2] A. Baeza, P. Mulet and D. Zoro, *High order accurate extrapolation technique for finite difference methods on complex domains with Cartesian meshes*, Journal of Scientific Computing, 66 (2016), 761-791.

[3] 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 (1995), 1241-1252.

[4] G. Chesshire and W.D. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, Journal of Computational Physics, 90 (1990), 1-64.

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

[6] M. Goldberg and E. Tadmor, *Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. I*, Mathematics of Computation, 32 (1978), 1097-1107.

[7] M. Goldberg and E. Tadmor, *Scheme-independent stability criteria for difference approximations of hyperbolic initial-boundary value problems. II*. Mathematics of Computation, 36 (1981), 603-626.

[8] B. Gustafsson and A. Sundström, *Incompletely parabolic problems in fluid dynamics*, SIAM Journal on Applied Mathematics, 35 (1978), 343-357.

[9] W.D. Henshaw and K.K. Chand, *A composite grid solver for conjugate heat transfer in fluid-structure systems*, Journal of Computational Physics, 228 (2009), 3708-3741.

[10] W.D. Henshaw H.-O. Kreiss and L.G. Reyna, *A fourth-order-accurate difference approximation for the incompressible Navier-Stokes equations*, Computers and fluids, 23 (1994), 575-593.

[11] L. Huang, C.-W. Shu and M. Zhang, *Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation*, Journal of Computational Mathematics, 26 (2008), 336-346.

[12] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1996), 202-228.

[13] Y. Jiang, C.-W. Shu and M. Zhang, *Free-stream preserving finite difference schemes on curvilinear meshes*, Methods and Applications of Analysis, 21 (2014), 1-30.

[14] H.-O. Kreiss and N.A. Petersson, *A second order accurate embedded boundary method for the wave equation with Dirichlet data*, SIAM Journal on Scientific Computing, 27 (2006), 1141-1167.

[15] H.-O. Kreiss, N.A. Petersson and J. Yström, *Difference approximations for the second order wave equation*, SIAM Journal on Numerical Analysis, 40 (2002), 1940-1967.

[16] H.-O. Kreiss, N. A. Petersson and J. Yström, *Difference approximations of the Neumann problem for the second order wave equation*, SIAM Journal on Numerical Analysis, 42 (2004), 1292-1323.

[17] 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 (2016), 140-158.

[18] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annual Review of Fluid Mechanics, 37 (2005), 239-261.

[19] S. Nilsson, N.A. Petersson, B. Sjögreen and H.-O. Kreiss, *Stable difference approximations for the elastic wave equation in second order formulation*, SIAM Journal on Numerical Analysis, 45 (2007), 1902-1936.

[20] J. Nordström, *Well posed problems and boundary conditions in computational fluid dynamics*, 22nd AIAA Computational Fluid Dynamics Conference, Dallas, Texas, USA, 2015.

[21] J. Nordström and M. Svörd, *Well-posed boundary conditions for the Navier–Stokes equations*, SIAM Journal on Numerical Analysis, 43 (2005), 1231-1255.

[22] S. Peskin, *The immersed boundary method*, Acta Numerica, 11 (2002), 1-39.

[23] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), 439-471.

[24] B. Sjögreen and N.A. Petersson, *A Cartesian embedded boundary method for hyperbolic conservation laws*, Communications in Computational Physics, 2 (2007), 1199-1219.

[25] K. Sebastian and C.-W. Shu, *Multidomain WENO finite difference method with interpolation at subdomain interfaces*, Journal of Scientific Computing, 19 (2003), 405-438.

[26] J.C. Strikwerda, *Initial boundary value problems for incompletely parabolic systems*, Communications on Pure and Applied Mathematics, 30 (1977), 797-822.

[27] C.-W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker and S. Osher, *High-order ENO schemes applied to two- and three-dimensional compressible flow*, Applied Numerical Mathematics, 9 (1992), 45-71.

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

[29] S. Tan and C.-W. Shu, *A high order moving boundary treatment for compressible inviscid flows*, Journal of Computational Physics, 230 (2011), 6023-6036.

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

[31] 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 (2012), 2510-2527.

[32] 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 (M^2AN), 49 (2015), 39-67.

[33] Y.-T. Zhang, J. Shi, C.-W. Shu and Y. Zhou, *Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high Reynolds numbers*, Physical Review E, 68 (2003), 046709.