Boundary treatment of linear multistep methods for hyperbolic conservation laws

Zuo, Hujian; Zhao, Weifeng; Lin, Ping

Published in:
Applied Mathematics and Computation

DOI:
10.1016/j.amc.2022.127079

Publication date:
2022

Boundary treatment of linear multistep methods for hyperbolic conservation laws

Hujian Zuo

Department of Applied Mathematics, University of Science and Technology Beijing, Beijing 100083, China

Weifeng Zhao*

Department of Applied Mathematics, University of Science and Technology Beijing, Beijing 100083, China

Ping Lin*

Division of Mathematics, University of Dundee, Dundee, DD1 4HN, Scotland, United Kingdom

Abstract

When using high-order schemes to solve hyperbolic conservation laws in bounded domains, it is necessary to properly treat boundary conditions so that the overall accuracy and stability are maintained. In [1, 2] a finite difference boundary treatment method is proposed for Runge-Kutta methods of hyperbolic conservation laws. The method combines an inverse Lax-Wendroff procedure and a WENO type extrapolation to achieve desired accuracy and stability. In this paper, we further develop the boundary treatment method for high-order linear multistep methods (LMMs) of hyperbolic conservation laws. We test the method through both 1D and 2D benchmark numerical examples for two third-order LMMs, one with a constant time step and the other with a variable time step. Numerical examples show expected high order accuracy and excellent stability. In addition, the approach in [3] may be adopted to deal with an exceptional case where eigenvalues of the flux Jacobian matrix change signs at the boundary. These results demonstrate that the combined boundary treatment

*Corresponding author

Email addresses: hjzuo@xs.ustb.edu.cn (Hujian Zuo), wfzhao@ustb.edu.cn (Weifeng Zhao), plin@maths.dundee.ac.uk (Ping Lin)

method works very well for LMMs of hyperbolic conservation laws.

Keywords: hyperbolic conservation laws; linear multistep methods; boundary treatment; inverse Lax-Wendroff

1. Introduction

When solving time dependent partial differential equations (PDEs), it is common to first discretize the spatial derivatives to form a system of ordinary differential equations (ODEs), and then solve the system of ODEs by time-stepping methods, such as Runge-Kutta methods and linear multistep methods (LMMs). This is the so-called method of lines (MOL) and is efficient for many time dependent PDEs [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. When the PDEs to be solved are restricted in frequently encountered bounded domains, it is necessary to properly treat boundary conditions so that the overall accuracy and stability are maintained. The boundary treatment is a challenge not only for the MOL but also for many other space-time discretizations [15, 16, 17, 18, 19, 20, 21, 22, 23].

This paper is concerned with the boundary treatment of MOL for hyperbolic conservation laws. For such equations, Tan and Shu [1] developed an efficient boundary treatment method on Cartesian meshes, where finite difference WENO schemes and the Runge-Kutta method are used for the spatial and time discretizations, respectively. The key point of this method is a so-called inverse Lax-Wendroff (ILW) procedure, which repeatedly utilizes the PDEs to convert spatial derivatives of inflow boundaries to time derivatives. For outflow boundaries, the spatial derivatives are approximated by extrapolations with the solutions at interior points. With these derivatives, the ghost point values outside the boundaries are then approximated by Taylor expansions. Since the algebra of the ILW procedure is very heavy for nonlinear PDEs, Tan *et al.* [2] simplify it by only using the ILW procedure for derivatives up to the first order and higher order derivatives are computed by extrapola-

tions. Numerical examples in [2] show that this simplification does not affect the stability and accuracy of the boundary treatment. Furthermore, Lu *et al.* [3] point out that the ILW procedure in [1, 2] will be much less robust if the eigenvalues of the Jacobian matrix of the flux are close to zero. As an improvement, it is suggested in [3] to compute the fluxes and the conservative variables separately at the ghost points. However, the boundary treatments in [1, 2, 3] all rely on the intermediate-stage boundary conditions [24], which are not available for Runge-Kutta methods of order higher than three. To adress this issue, a new boundary treatment method is proposed in [25] by using the Runge-Kutta method itself rather than the intermediate-stage boundary conditions at the boundaries. This technique, combined with the ILW procedure, possesses designed accuracy and good stability and applies to both explicit and implicit-explicit Runge-Kutta methods of arbitrary orders [25, 26]. On the other hand, the ILW procedure has also been adapted to convection-diffusion equations [27] and the Boltzmann equation [28]. The stability of the ILW procedure applied to linear diffusion and hyperbolic equations is analyzed in [29, 30]. Other developments and applications of the procedure can be found in [31, 32, 33, 34].

Note that all the boundary treatment methods mentioned above are proposed for Runge-Kutta methods. In this paper, we consider the boundary treatment of LMMs accompanying finite difference WENO schemes for hyperbolic conservation laws. Since there are no intermediate stages in LMMs, the boundary treatment method in [2] can be applied without considering the intermediate stages. Specifically, we use the ILW procedure to deal with the inflow boundary conditions to obtain the solutions and their first order derivatives at the boundary, while high order derivatives and the outflow boundary conditions are extrapolated. Here we use Lagrange extrapolations when the solution is smooth and otherwise WENO type extrapolations in [2] are used to avoid oscillations. Moreover, the approach in [3] is employed

to deal with the case where the eigenvalues of the Jacobi matrix of the flux change signs at the boundary. We test the method for two third-order LMMs, *i.e.*, an Adams method with constant time step and a variable step-size backward-differentiation-formula (VSSBDF) method. Numerical examples in both one and two dimensions validate the designed accuracy and good stability of the boundary treatment method for the two LMMs. Our results demonstrate that the combined boundary treatment method works well for LMMs of hyperbolic conservation laws.

This paper is organized as follows. In Section 2, we introduce the LMMs for hyperbolic conservation laws. In Section 3, we introduce the boundary treatment method by means of a one dimension system. Section 4 presents the numerical experiments and finally Section 5 concludes the paper.

2. Scheme formulation

Consider a one-dimensional (1D) hyperbolic system

$$U_t + F(U)_x = 0, \tag{2.1}$$

in the domain $x \in (0,1)$, where $U = U(x,t) \in \mathbb{R}^M$ is the unknown variable with initial data $U(x,0) = U_0(x)$ and $F(U)$ is the flux. Denote by $F_U(U)$ the Jacobian matrix of $F(U)$ and assume that $F_U(U(0,t))$ has p positive eigenvalues. Then p independent boundary conditions are needed at the left boundary $x = 0$, which are assumed to be

$$B(U(0,t),t) = 0. \tag{2.2}$$

Similarly, appropriate boundary conditions are imposed at the right boundary $x = 1$.

We use the finite difference WENO scheme with a uniform mesh size Δx to solve (2.1). Denote by x_j the j-th grid point and by U_j the numerical solution of U at x_j. The finite

difference WENO scheme [35] for the spatial derivative $F(U)_x$ at x_j takes a conservative form

$$\frac{\hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}}}{\Delta x}.$$

Here $\hat{F}_{j+\frac{1}{2}}$ is the numerical flux at the interface $x_{j+\frac{1}{2}} = (x_j + x_{j+1})/2$ and we adopt the global Lax-Friedrichs splitting flux throughout the paper. After the spatial discretization, we obtain a semi-discrete form

$$\frac{dU_j}{dt} = L(U_j) := -\frac{\hat{F}_{j+\frac{1}{2}} - \hat{F}_{j-\frac{1}{2}}}{\Delta x}. \tag{2.3}$$

For the above semi-discrete form as a system of ODEs, we solve it with a general k-step LMM

$$\sum_{i=0}^{k} \alpha_{i,n} U_j^{n+i} = \Delta t_{n+k-1} \sum_{i=0}^{k} \beta_{i,n} L(U_j^{n+i}), \tag{2.4}$$

where U_j^n is the numerical solution of U at position x_j and time t_n, $\Delta t_{n+k} = t_{n+k+1} - t_{n+k}$ is the time step, and $\alpha_{i,n}$ and $\beta_{i,n}$ are the LMM coefficients. Note that the time step Δt_n can be a constant or a variable changing with the level n. When Δt_n is a constant, the coefficients $\alpha_{i,n}$ and $\beta_{i,n}$ are constants and satisfy [36]

$$\alpha_{k,n} = 1, \ |\alpha_{0,n}| + |\beta_{0,n}| \neq 0.$$

When Δt_n is a variable, the coefficients $\alpha_{i,n}$ and $\beta_{i,n}$ are functions of step size ratios $\omega_i = \frac{\Delta t_i}{\Delta t_{i-1}}, i = n+1, n+2, ..., n+k-1$. The order conditions on these coefficients for the LMM (2.4) to be q-th order accurate are [37]

$$\begin{aligned}
\sum_{i=0}^{k} \alpha_{i,n} &= 0, \\
\sum_{i=1}^{k} \alpha_{i,n} \left(\sum_{j=0}^{i-1} \Delta t_{n+j} \right) &= \Delta t_{n+k-1} \left(\sum_{i=0}^{k} \beta_{i,n} \right), \\
&\vdots \\
\frac{1}{q!} \sum_{i=1}^{k} \alpha_{i,n} \left(\sum_{j=0}^{i-1} \Delta t_{n+j} \right)^q &= \frac{\Delta t_{n+k-1}}{(q-1)!} \sum_{i=1}^{k} \beta_{i,n} \left(\sum_{j=0}^{i-1} \Delta t_{n+j} \right)^{q-1}.
\end{aligned}$$

In this paper, we consider two LMMs, *i.e.*, a constant time step method and a variable time step one. Specifically, the constant time step method we adopt is the third-order explicit Adams method [38]

$$U_j^{n+1} = U_j^n + \frac{\Delta t}{12}\left[23L(U_j^n) - 16L(U_j^{n-1}) + 5L(U_j^{n-2})\right]. \tag{2.5}$$

The variable time step method we consider is the third-order variable step-size backward-differentiation-formula (VSSBDF3) method proposed in [37]:

$$\sum_{i=0}^{3} \alpha_{i,n} U_j^{n+i} = \Delta t_{n+2} \sum_{i=0}^{2} \beta_{i,n} L(U_j^{n+i}), \tag{2.6}$$

where

$$\begin{aligned}
\alpha_{0,n} &= -\frac{\omega_{n+1}^3 \omega_{n+2}^2 (1+\omega_{n+2})}{(1+\omega_{n+1})(1+\omega_{n+1}+\omega_{n+1}\omega_{n+2})},\\
\alpha_{1,n} &= \omega_{n+2}^2 \left(\omega_{n+1} + \frac{1}{1+\omega_{n+2}}\right),\\
\alpha_{2,n} &= -1 - \omega_{n+2} - \frac{\omega_{n+1}\omega_{n+2}(1+\omega_{n+2})}{1+\omega_{n+1}},\\
\alpha_{3,n} &= 1 + \frac{\omega_{n+2}}{1+\omega_{n+2}} + \frac{\omega_{n+1}\omega_{n+2}}{1+\omega_{n+1}(1+\omega_{n+2})},\\
\beta_{0,n} &= \frac{\omega_{n+1}^2 \omega_{n+2}(1+\omega_{n+2})}{1+\omega_{n+1}},\\
\beta_{1,n} &= -\omega_{n+2}[1+\omega_{n+1}(1+\omega_{n+2})],\\
\beta_{2,n} &= \frac{(1+\omega_{n+2})[1+\omega_{n+1}(1+\omega_{n+2})]}{1+\omega_{n+1}}.
\end{aligned}$$

3. Boundary treatment

In this section, we extend the idea in [2] to develop a boundary treatment method for LMMs, using the above LMMs (as illustrative examples) accompanying finite difference WENO schemes. We note that unlike Runge-Kutta methods, no intermediate stages and their treatment are needed in LMMs. Additionally, the approach in [3] can be combined to deal with the case when there exists zero eigenvalue at the boundary, which will cause

instability for the method in [2] (see Remark 3.1). To be concrete, we employ the third-order WENO scheme and take $\Delta t = O(\Delta x)$.

3.1. Computation of U and $F(U)$ at ghost points

We consider the implemention at the left boundary $x = 0$ and the right boundary $x = 1$ can be treated in a similar way. Without loss of generality, we discretize the domain $(0, 1)$ by a uniform mesh

$$\frac{\Delta x}{2} = x_0 < x_1 < x_2 < \cdots < x_N = 1 - \frac{\Delta x}{2}.$$

For the third-order WENO scheme, two ghost points $x_{-1} = x_0 - \Delta x$ and $x_{-2} = x_0 - 2\Delta x$ outside the boundary are needed. Assume that the interior solutions $U_j, j = 0, 1, 2, \ldots, N$, have been updated from time level t_{n-1} to time level t_n. Since the spatial discretization is third-order, we use a third-order Taylor approximation to construct the values at the ghost points,

$$U_j^n = \sum_{k=0}^{2} \frac{x_j^k}{k!} U^{n,(k)}, \quad j = -1, -2, \tag{3.1}$$

where $U^{n,(k)}$ denote the approximations of the k-th order derivative of the spatial derivative at the boundary point $\frac{\partial^k U}{\partial x^k}\big|_{x=0,t=t_n}$. Once $U^{n,(k)}, k = 0, 1, 2$, are provided, U_j^n at the ghost points can be obtained. Then the flux is directly computed by

$$(F(U))_j^n = F(U_j^n), \quad j = -1, -2. \tag{3.2}$$

3.2. Computation of $U^{n,(k)}$

3.2.1. $k = 0$

We first do a local characteristic decomposition to determine incoming and outgoing characteristic variables at the boundary as in [2]. Denote the Jacobian matrix of the flux evaluated at $x = x_0$ by

$$A(U_0^n) = \partial_U F(U)\big|_{U=U_0^n},$$

and assume that it has p positive eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_p$ and $(M-p)$ negative eigenvalues $\lambda_{p+1}, \lambda_{p+2}, \ldots, \lambda_M$ with $l_1, l_2, ..., l_p$ and $l_{p+1}, l_{p+2}, ..., l_M$ the corresponding left eigenvectors, respectively. Define by $V_{j,m}$ the m-th component of the local characteristic variable at grid point $x_j, j = 0, 1, 2$, *i.e.*,

$$V_{j,m} = l_m U_j^n, \quad m = 1, 2, \ldots, M, \quad j = 0, 1, 2. \tag{3.3}$$

We extrapolate the outgoing characteristic variable $V_{j,m}, j = 0, 1, 2, m = p+1, p+2, \ldots, M$ to the boundary x_b =0 (for smooth solutions, we use Lagrangian extrapolation, otherwise the WENO type extrapolation in [1, 2] can be used to avoid possible oscillations), and denote the extrapolated k-th order derivative by

$$V_{x_b,m}^{*(k)}, \quad k = 0, 1, 2. \tag{3.4}$$

With $V_{x_b,m}^{*(0)}$ and the boundary condition (2.2), $U^{n,(0)}$ at the boundary can be determined from the following equations

$$\begin{aligned} &l_m U^{n,(0)} = V_{x_b,m}^{*(0)}, \quad m = p+1, p+2, \ldots, M, \\ &B(U^{n,(0)}, t_n) = 0. \end{aligned} \tag{3.5}$$

3.2.2. $k = 1$

Having $U^{n,(0)}$, we proceed to compute $U^{n,(1)}$ with the ILW procedure proposed in [1, 2]. To do this, we differentiate the boundary condition (2.2) with respect to t

$$B_U(U^{n,(0)}, t_n)\partial_t U^{n,(0)} + B_t(U^{n,(0)}, t_n) = 0,$$

which can be written as

$$B_U(U^{n,(0)}, t_n)\partial_t U^{n,(0)} = g(U^{n,(0)}, t_n),$$

with $g(U^{n,(0)}, t_n) := -B_t(U^{n,(0)}, t_n)$. In addition, multiplying the equation (2.1) with $B_U(U^{n,(0)}, t_n)$ from the left yields

$$B_U(U^{n,(0)}, t_n)\partial_t U^{n,(0)} + B_U(U^{n,(0)}, t_n)A(U^{n,(0)})U^{n,(1)} = 0.$$

With the above two equations and $V^{*(1)}_{x_b,m}$ obtained by the extrapolation, $U^{n,(1)}$ can be determined by solving

$$l_m U^{n,(1)} = V^{*(1)}_{x_b,m}, \quad m = p+1, p+2, \ldots, M, \tag{3.6a}$$

$$B_U(U^{n,(0)}, t_n)A(U^{n,(0)})U^{n,(1)} = -g(U^{n,(0)}, t_n). \tag{3.6b}$$

This is the ILW procedure in [1, 2].

3.2.3. $k = 2$

Following [2], we simply extrapolate $\frac{\partial^k U^n}{\partial x^k}|_{x=0}$ for $k = 2$, a procedure that will not affect the stability [2]. First, note that we have already obtained the characteristic variables at grid points near the boundary in (3.3). Based on these characteristic variables, Lagrangian extrapolation (for smooth solutions) or the WENO type extrapolation (for nonsmooth solutions) is employed to compute the derivatives $V^{*(2)}_{x_b,m}$ for each $m = 1, 2, \ldots, M$. Following this step, the approximation of $\frac{\partial^2 U^n}{\partial x^2}|_{x=0}$ is given by

$$U^{n,(2)} = l^{-1}V^{*(2)}_{x_b,m}, \tag{3.7}$$

where l is the local left eigenvector matrix composed of $l_1, l_2, \ldots, l_M$.

Remark 3.1. *As pointed out in [3], when an eigenvalue of $F_U(U)$ at the boundary is close to zero or even varies its sign with t, the coefficient matrix of the system* (3.6) *will be very stiff and thus leads to numerical instability. To address this issue, it is suggested in [3] to directly approximate the flux $F(U)$ at ghost points instead of approximating U first and then computing*

the flux as (3.2). *This improvement perfectly avoids the above instability and will also be adopted in our numerical examples. In our simulations, we choose the boundary treatment method a priory and globally. When there are exact solutions to the problems simulated, e.g. the vortex evolution problem, we use the solutions to determine which boundary treatment to employ. For the double Mach reflection problem containing zero eigenvalue at the boundary due to the solid wall condition, we use the boundary treatment in [3] directly. If the signs of the eigenvalues at the boundary can not be judged a priory, one may adaptively choose the boundary treatment method or entirely use the boundary treatment in [3].*

Remark 3.2. *For a k-step LMM, initial data at the first k time steps are required. In 1D numerical examples and the 2D double Mach reflection example, we take these initial data by using the Runge-Kutta methods together with the boundary treatment in [25, 26]. For the 2D smooth examples we compute, the initial data are determined by the analytical solution for simplicity.*

Remark 3.3. *Here we only introduce the boundary treatment for 1D system and it can be extended to 2D case as in [2]. See [2] for more details. In our numerical examples, the 2D boundary treatment has been tested as well.*

4. Numerical examples

In this section, we conduct several numerical experiments in both one and two dimensions (most of the examples are from [1]) to verify the above boundary treatment. With the third-order finite difference WENO scheme for spatial discretization, we consider two third-order LMMs. Specifically, we apply the explicit third-order Adams method (2.5) from subsection 4.1 to subsection 4.3 and employ the VSSBDF3 method (2.6) in subsection 4.4 and subsection 4.5.

4.1. 1D linear equation

We start with a linear scalar hyperbolic equation

$$u_t + u_x = 0, \tag{4.1}$$

with smooth and discontinuous solutions. We first consider a smooth solution $u = e^{x-t}$ in the domain $(0, 1)$. To match this solution, the initial data and the boundary condition are prescribed as

$$u(x, 0) = e^x, \quad u(0, t) = e^{-t}. \tag{4.2}$$

We take $\Delta t = 0.4\Delta x$ and compute l^1, l^2 and l^∞ errors at the terminal time $t = 0.5$. The errors in Table 4.1 show that the convergence order is about three, which is the designed order.

Table 4.1: Error table for the problem (4.1) with smooth solution.

Δx	l^1 error	order	l^2 error	order	l^∞ error	order
1/20	9.79e-6		1.79e-5		7.23e-5	
1/40	1.18e-6	3.06	1.94e-6	3.20	9.58e-6	2.92
1/80	1.47e-7	3.00	2.22e-7	3.13	1.23e-6	2.96
1/160	1.90e-8	2.95	2.72e-8	3.03	1.58e-7	2.96
1/320	2.65e-9	2.85	3.64e-9	2.90	2.07e-8	2.94
1/640	3.34e-10	2.99	4.60e-10	2.98	2.68e-9	2.95

Next we consider the problem from [1] with discontinuous solution to (4.1). The initial data is $u(0, x) = 0.25 + 0.5\sin(\pi x), x \in [-1, 1]$ and the boundary condition is taken as

$$u(-1, t) = \begin{cases} 0.25, t \leq 1, \\ \\ -1, t > 1. \end{cases} \tag{4.3}$$

The exact solution of this problem is [1]

$$u(x, t) = \begin{cases} -1, x < t - 2, \\ \\ 0.25, t - 2 \leq x < t - 1, \\ \\ 0.25 + 0.5\sin[\pi(x - t)], x \geq t - 1. \end{cases} \tag{4.4}$$

According to the solution, there is a removable discontinuity point at $x = t - 1$ and a jump discontinuity point at $x = t - 2$ when $1 < t < 3$. In the computation, we take $\Delta t = 0.5\Delta x$ with $\Delta x = 1/80$. Numerical solutions at $t = 1.2$ and $t = 2.2$ are shown in Figure 4.1. It can be seen that both types of discontinuities are well captured.

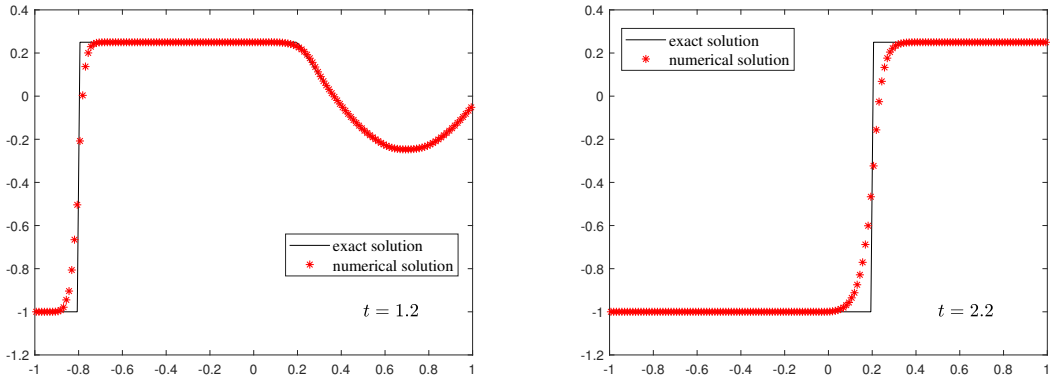

Figure 4.1: Numerical solutions of the equation (4.1) with boundary condition (4.3).

4.2. 1D Burgers equation

Next we consider the nonlinear Burgers equation with source term

$$u_t + \left(\frac{u^2}{2}\right)_x = u^2 - u, \tag{4.5}$$

in domain $x \in (0, 1)$. It is easy to verify that $u(x, t) = e^{x-t}$ is an exact solution to this equation. Since $f'(u) = u > 0$, one boundary condition needs to be imposed at $x = 0$. Similar to the previous example, to match the exact solution we set the boundary condition and the initial data as

$$u(0, t) = e^{-t}, \quad u(x, 0) = e^{x}.$$

With fixed $\Delta t = 0.2\Delta x$, the errors at $t = 1$ are given in Table 4.2. We see that designed third-order convergence is achieved again for the nonlinear equation.

Table 4.2: Error table for the Burgers equation (4.5) with smooth solution.

Δx	l^1 error	order	l^2 error	order	l^∞ error	order
1/20	1.01e-5		1.94e-5		8.02e-5	
1/40	1.09e-6	3.21	1.97e-6	3.30	1.05e-5	2.93
1/80	1.29e-7	3.07	2.10e-7	3.23	1.35e-6	2.96
1/160	1.63e-8	2.99	2.42e-8	3.12	1.73e-7	2.97
1/320	2.25e-9	2.85	3.15e-9	2.95	2.24e-8	2.94
1/640	2.96e-10	2.93	4.01e-10	2.97	2.89e-9	2.96

4.3. 1D linear system

We next test the boundary treatment with 1D linear system

$$U_t - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} U_x = 0, \tag{4.6}$$

where $U = (u, v)^T$. This system admits an exact solution

$$U = \begin{pmatrix} \sin x \cos t \\ \cos x \sin t \end{pmatrix}. \tag{4.7}$$

Since the Jacobian matrix has two eigenvalues 1 and -1, one boundary condition is needed at $x = 0$ and $x = 1$, respectively. We impose them as

$$u(0, t) = 0,$$

$$v(1, t) = \cos 1 \sin t.$$

Table 4.3 gives the errors at $t = 0.3$ with $\Delta t = 0.6\Delta x$, from which third-order convergence can be observed.

Table 4.3: Error table for the problem (4.6) with solution (4.7).

Δx	l^1 error	order	l^2 error	order	l^∞ error	order
1/20	1.74e-5		1.89e-5		2.93e-5	
1/40	2.16e-6	3.01	2.31e-6	3.03	3.93e-6	2.90
1/80	2.58e-7	3.06	2.75e-7	3.07	5.04e-7	2.96
1/160	2.75e-8	3.23	3.00e-8	3.20	5.90e-8	3.10
1/320	2.62e-9	3.39	2.91e-9	3.37	5.23e-9	3.49
1/640	3.39e-10	2.95	3.63e-10	3.00	6.26e-10	3.06

4.4. 1D Euler equations

This example is also from [1] and it is devoted to testing the boundary treatment method for nonlinear systems. Specifically, we solve the Euler equations

$$\begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u\left(E + p\right) \end{pmatrix}_x = 0, \tag{4.8}$$

where ρ is the fluid density, u is the velocity, p is the pressure, and E is the total energy given by

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

with $\gamma = 1.4$ the specific heat ratio.

4.4.1. Smooth solution

We first test the accuracy with the analytical solution

$$\begin{cases} \rho\left(x,t\right) = 1 + 0.2\sin\left(x - t\right), \\ u\left(x,t\right) = 1, \\ p\left(x,t\right) = 2, \end{cases} \tag{4.9}$$

in the domain $x \in (-\pi, \pi)$. Since the Jacobian matrix has one negative eigenvalue and two positive ones at $x = \pi$ and $x = -\pi$, respectively, one and two boundary conditions are needed at $x = \pi$ and $x = -\pi$, respectively. We use the analytical solution to prescribe the boundary conditions as

$$\begin{cases} \rho\left(-\pi,t\right) = 1 + 0.2\sin t, \\ u\left(-\pi,t\right) = 1, \\ \rho\left(\pi,t\right) = 1 + 0.2\sin t. \end{cases}$$

We take $\Delta t = CFL\frac{\Delta x}{\max\{|\lambda_i|\}}$ with $CFL = 0.4$ and λ_i the i-th eigenvalue of $F_U(U)$. The errors at $t = 0.3$ are listed in Table 4.4. Clearly, designed third-order convergence order is achieved.

Table 4.4: Error table for the problem (4.8) with solution (4.9).

Δx	l^1 error	order	l^2 error	order	l^∞ error	order
$\pi/80$	1.19e-5		2.12e-5		8.47e-5	
$\pi/160$	6.79e-7	4.13	9.98e-7	4.41	3.14e-6	4.75
$\pi/320$	3.67e-8	4.21	4.83e-8	4.37	1.09e-7	4.85
$\pi/640$	2.25e-9	4.03	3.46e-9	3.80	1.24e-8	3.13
$\pi/1280$	2.28e-10	3.30	3.87e-10	3.16	1.56e-9	2.99
$\pi/2560$	2.85e-11	3.00	4.79e-11	3.01	1.96e-10	3.00

4.4.2. Discontinuous solution

Next we simulate the interaction of two blast waves [39] with the initial condition

$$\rho = 1, \quad u = 0, \quad p = \begin{cases} 1000, & 0 < x < 0.1, \\ 0.01, & 0.1 < x < 0.9, \\ 100, & 0.9 < x < 1. \end{cases} \tag{4.10}$$

The computational domain is $(0, 1)$ and there are solid walls at $x = 0$ and $x = 1$. This problem involves multiple interactions of shocks and rarefactions with each other and with contact discontinuities. We use the boundary treatment at both walls and the solutions at the first three time levels are computed by RK methods together with the boundary treatment in [25, 26]. The density profiles at $t = 0.038$ with $\Delta x = 1/800$ and $\Delta x = 1/1600$ are plotted in Figure 4.2. It is clear that the computational results are convergent as the mesh is refined. Additionally, they agree well with the reference solution, which is obtained by the fifth-order WENO scheme with the reflection technique for the boundary treatment. These demonstrate the good performance of the boundary treatment for problems with shocks.

4.4.3. Stability test

We further test the stability of the boundary treatment method for the Euler equations with the solution (4.9) by fixing $\Delta x = \pi/80$ and varying the CFL number. The solutions of the first three time levels for the LMM are obtained by RK methods with the boundary

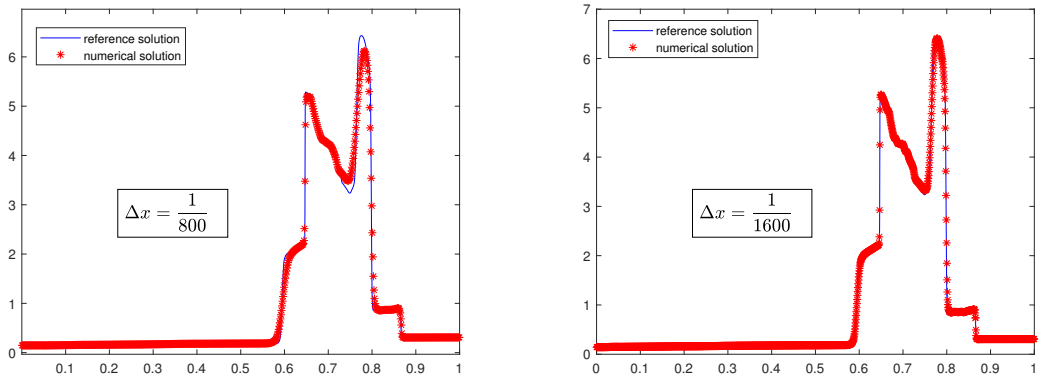

Figure 4.2: Computational results of the interaction of two blast waves. The reference solution is computed by the fifth-order WENO scheme with the reflection technique for the boundary treatment and with $\Delta x = 1/3200$.

treatment in [25, 26] or from the analytical solution (4.9). We compare the results of the present boundary treatment with those obtained with the periodic boundary conditions. The step size of the CFL number is taken as 0.01 and we stop the computation when $\log(l^1 \text{ error}) > 1.5$. The l^1 errors as a function of the CFL number in different cases are shown in Figure 4.3. It can be seen that if the analytical solution is used for the first three time levels, the errors of the boundary treatment are almost the same with those of the periodic boundary conditions. Surprisingly, if the solutions at the first three time levels are computed by the RK methods, then the presented boundary treatment possesses better stability and smaller errors than the periodic boundary conditions. These results demonstrate the good stability of the present boundary treatment method.

4.5. 2D Euler equations

In this section we test the boundary treatment for the 2D Euler equations

$$\partial_t U + \partial_x F(U) + \partial_y G(U) = 0, \quad (x, y) \in \Omega \tag{4.11}$$

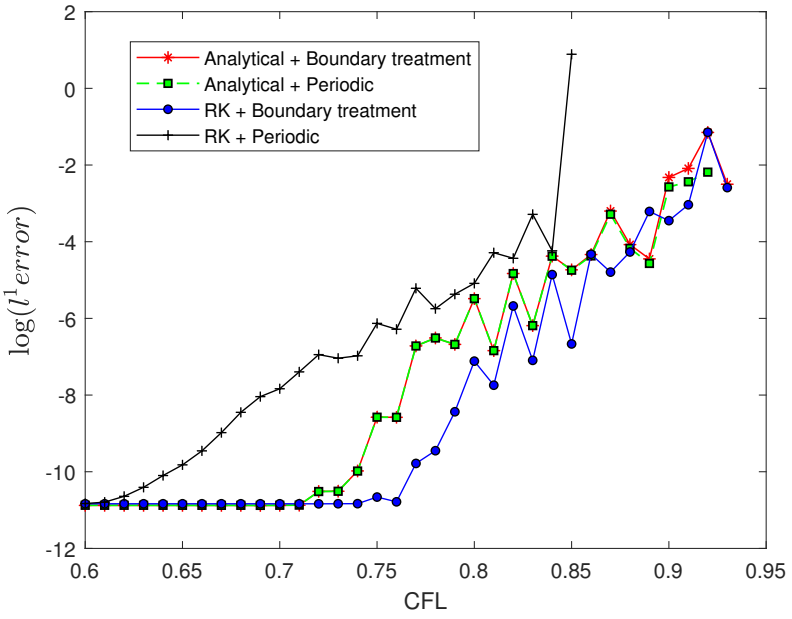

Figure 4.3: l^1 errors as a function of the CFL number for the Euler equations (4.8) with analytical solution (4.9). "Analytical" and "RK" mean that solutions at the first three time levels are given by the analytical solution (4.9) and the Runge-Kutta methods, respectively. "Periodic" and "Boundary treatment" mean that the ghost point solutions are obtained by periodic boundary conditions and the present boundary treatment, respectively.

with

$$U = (\rho, \rho u, \rho v, E)^T,$$

$$F(U) = (\rho u, \rho u^2 + p, \rho uv, (E + p)u)^T,$$

$$G(U) = (\rho v, \rho uv, \rho v^2 + p, (E + p)v)^T,$$

and equation of state $E = \frac{1}{2}\rho(u^2 + v^2) + \frac{p}{\gamma - 1}$. Here ρ is the density, u and v are the velocities in x and y directions, E is the total energy, p is the pressure, and $\gamma = 1.4$ is the specific heat ratio.

4.5.1. Smooth solution

First, we simulate the vortex problem with analytical solutions to the above 2D Euler equations as in [1]. In this problem, the mean flow $\rho = p = u = v = 1$ is initially imposed with an isentropic vortex perturbation centered at (x_0, y_0) in (u, v) and in the temperature

$T = p/\rho$, no perturbation in the entropy $S = p/\rho^\gamma$:

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}),$$
$$\delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{(1-r^2)},$$
$$\delta S = 0.$$

Here $(\bar{x}, \bar{y}) = (x - x_0, y - y_0)$, $r^2 = \bar{x}^2 + \bar{y}^2$ and ϵ is the vortex strength. The exact solution $U(t, x, y)$ of this problem is a passive convection of the vortex with the mean velocity, *i.e.*, $U(t, x, y) = U(0, x - ut, y - vt)$. As in [1], we set the spatial domain as a square $\Omega = [-0.5, 1]^2$ or a disk $\Omega = \{(x, y) : x^2 + y^2 < 0.5^2\}$ with $(x_0, y_0) = (0, 0)$, and take boundary conditions from the exact solution whenever needed.

We consider two cases, $\varepsilon = 1$ and $\varepsilon = 5$. It is direct to verify that when $\varepsilon = 1$ the signs of eigenvalues at the boundary remain unchanged while when $\varepsilon = 5$ the signs vary before the terminal time 0.01. For the second case with varying signs we adopt the method in [3] for the boundary treatment (see also Remark 3.1). We solve the two cases on both square and disk domains. In the computation we set $CFL = 0.3$ and the solutions at the first three time levels are determined by the exact solution above. The errors are listed in Tables 4.5 and 4.6, from which third-order convergence can be observed for square and disk domains. These verify the accuracy of the boundary treatment method for straight and curved boundaries in 2D.

4.5.2. Discontinuous solution

Finally, we compute the double Mach reflection problem [39, 1] involving a reflecting wall to test the boundary treatment method for discontinuous solutions of the Euler equations. This problem describes a shock which moves horizontally into a solid wall inclined by a 30° angle. The computational domain is the same as in [1] and the solid wall boundary condition is $(u, v) \cdot \boldsymbol{n} = 0$ with $\boldsymbol{n}$ the unit normal vector at the wall. Usually, people consider an

Table 4.5: Error table for the 2D Euler equations with $\varepsilon = 1$ and $\varepsilon = 5$ on the square. Here $\Delta y = \Delta x$.

			$\varepsilon = 1$			
Δx	l^1 error	order	l^2 error	order	l^∞ error	order
3/160	9.80e-9		3.11e-8		3.80e-7	
3/320	1.12e-9	3.13	4.20e-9	2.89	6.58e-8	2.53
3/640	1.35e-10	3.05	5.58e-10	2.91	1.05e-8	2.65
3/1280	1.58e-11	3.10	7.26e-11	2.94	2.42e-9	2.12
			$\varepsilon = 5$			
Δx	l^1 error	order	l^2 error	order	l^∞ error	order
3/160	6.17e-7		1.10e-6		1.06e-5	
3/320	3.98e-8	4.07	6.90e-8	3.99	2.29e-6	2.21
3/640	2.62e-9	3.92	7.23e-9	3.25	5.62e-7	2.03
3/1280	2.45e-10	3.42	1.02e-9	2.83	1.43e-7	1.97

Table 4.6: Error table for the 2D Euler equations with $\varepsilon = 1$ and $\varepsilon = 5$ on the disk. Here $\Delta y = \Delta x$.

			$\varepsilon = 1$			
Δx	l^1 error	order	l^2 error	order	l^∞ error	order
1/20	3.21e-7		8.79e-7		6.54e-6	
1/40	1.94e-7	0.72	7.26e-7	0.28	6.05e-6	0.11
1/80	4.08e-8	2.25	1.87e-7	1.96	2.57e-6	1.24
1/160	5.34e-9	2.93	2.65e-8	2.82	4.39e-7	2.55
1/320	7.99e-10	2.74	4.54e-9	2.55	1.00e-7	2.14
			$\varepsilon = 5$			
Δx	l^1 error	order	l^2 error	order	l^∞ error	order
1/20	1.02e-5		1.44e-5		7.26e-5	
1/40	2.98e-6	1.78	4.41e-6	1.71	3.04e-5	1.26
1/80	3.24e-7	3.20	8.06e-7	2.45	9.04e-6	1.75
1/160	3.41e-8	3.25	1.21e-7	2.73	1.62e-6	2.48
1/320	4.27e-9	3.00	2.06e-8	2.56	3.58e-7	2.18

equivalent problem that the solid wall is put at the bottom of computational domain and the shock makes a 60° angle with the wall. In the equivalent problem, the wall is parallel to the x axis so that one can solve the problem conveniently by the reflection technique. Here we simulate both the equivalent and the original problems as in [1].

We first solve the equivalent problem with boundary aligned with the grid lines and use our boundary treatment for the solid wall boundary condition. To demonstrate the effectiveness of our method, we compare the results with those of the reflection technique, which are shown in Figure 4.4. Clearly, the results from the two methods agree well.

For the original problem with wall not aligned with the grid lines, the present boundary treatment also performs well and the results are plotted in 4.5. We see that the results are similar to those of the equivalent problem in 4.4. These results demonstrate the effectiveness of the present boundary treatment method for discontinuous solutions with boundary aligned/not aligned with the grid lines.

5. Conclusions

In this paper, we consider the boundary treatment of LMMs accompanying finite difference WENO schemes for hyperbolic conservation laws. We extend the idea for Runge-Kutta methods in [2] to develop a boundary treatment method for LMMs. Unlike Runge-Kutta methods, no intermediate stages and their treatment [2] are needed for LMMs. Specifically, we use the ILW procedure to deal with the inflow boundary conditions to obtain the solutions and their first order derivatives at the boundary, while high order derivatives and the outflow boundary conditions are extrapolated. Here we use Lagrange extrapolations when the solution is smooth and otherwise WENO type extrapolations are used to avoid oscillations. Moreover, the approach in [3] is employed to deal with the case where the eigenvalues of

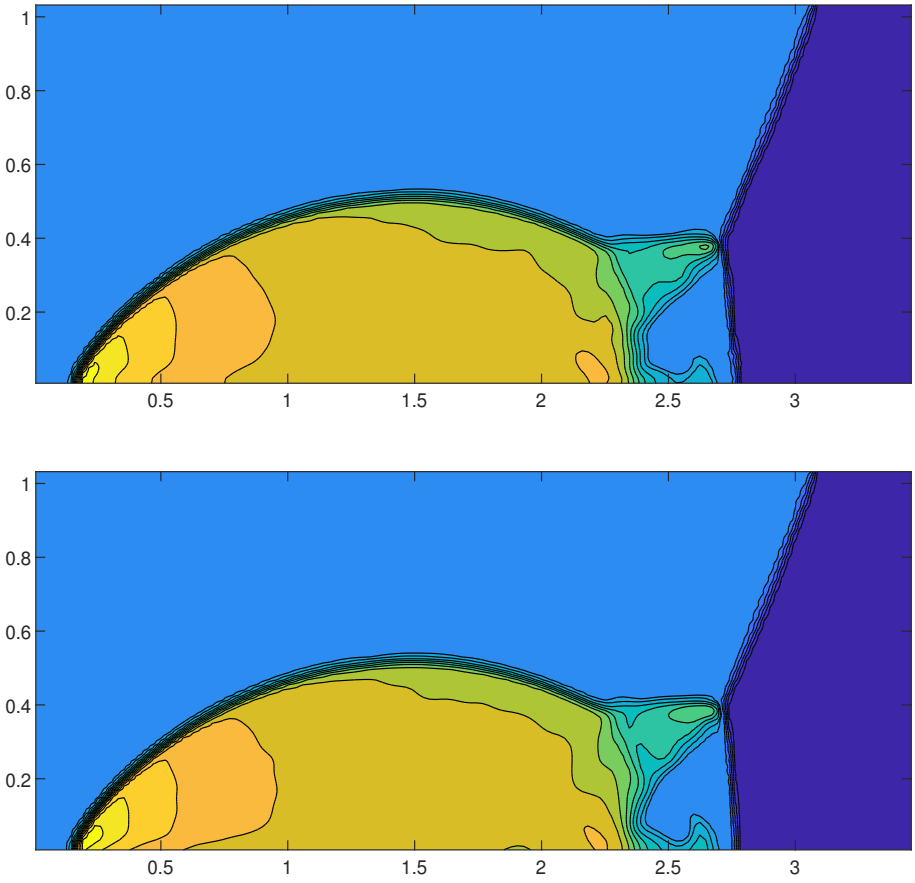

Figure 4.4: Density contours of the equivalent double Mach problem at $t = 0.2$ with $\Delta x = \sqrt{3}/120$ and 18 contour lines from 1.2 to 20. Top: present boundary treatment, bottom: the reflection technique.

the Jacobi matrix change signs at the boundary. We test the method for two third-order LMMs, *i.e.*, the Adams method with constant time step and the VSSBDF method. Numerical examples in one and two dimensions validate the designed accuracy and good stability of the boundary treatment method. Our results demonstrate that the combined boundary treatment method mentioned above works very well for LMMs of hyperbolic conservation laws.

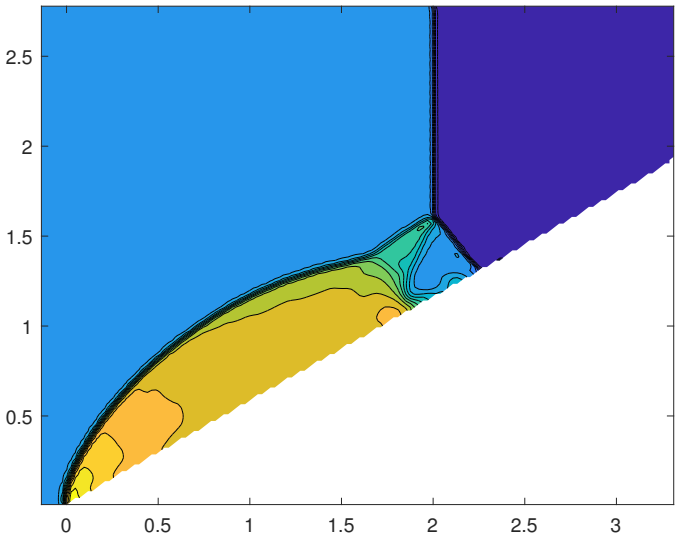

Figure 4.5: Density contours of the original double Mach problem at $t = 0.2$ with $\Delta x = \sqrt{3}/120$ and 18 contour lines from 1.2 to 20.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11771040, 11801030 and 11861131004) and the Fundamental Research Funds for the Central Universities (No. 06500073).

References

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

[2] Sirui Tan, Cheng Wang, Chi-Wang Shu, and Jianguo Ning. Efficient implementation of high order inverse Lax–Wendroff boundary treatment for conservation laws. *Journal of Computational Physics*, 231(6):2510–2527, 2012.

[3] Jianfang Lu, Chi-Wang Shu, Sirui Tan, and Mengping Zhang. An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary. *Journal of Computational Physics*, 426:109940, 2021.

[4] Xinghui Zhong and Chi-Wang Shu. Numerical resolution of discontinuous Galerkin methods for time dependent wave equations. *Computer Methods in Applied Mechanics and Engineering*, 200(41-44):2814–2827, 2011.

[5] Alessandra Nigro, Carmine De Bartolo, Andrea Crivellini, and Francesco Bassi. Second derivative time integration methods for discontinuous Galerkin solutions of unsteady compressible flows. *Journal of Computational Physics*, 350:493–517, 2017.

[6] Dietmar Kröner, Michael Růžička, and Ioannis Toulopoulos. Local discontinuous Galerkin numerical solutions of non-Newtonian incompressible flows modeled by p-Navier–Stokes equations. *Journal of Computational Physics*, 270:182–202, 2014.

[7] Gianmarco Manzini, Gianluca Maguolo, and Mario Putti. The high-order mixed mimetic finite difference method for time-dependent diffusion problems. *Journal of Scientific Computing*, 80(3):1805–1830, 2019.

[8] Dylan Abrahamsen and Bengt Fornberg. Explicit time stepping of PDEs with local refinement in space-time. *Journal of Scientific Computing*, 81(3):1945–1962, 2019.

[9] Lulu Tian, Hui Guo, Rui Jia, and Yang Yang. An $h-$adaptive local discontinuous Galerkin method for simulating wormhole propagation with Darcy–Forcheiner model. *Journal of Scientific Computing*, 82:43, 2020.

[10] Sigal Gottlieb and Steven J. Ruuth. Optimal strong-stability-preserving time-stepping

schemes with fast downwind spatial discretizations. *Journal of Scientific Computing*, 27(1-3):289–303, 2006.

[11] David I. Ketcheson, Colin B. MacDonald, and Steven J. Ruuth. Spatially partitioned embedded Runge–Kutta methods. *SIAM Journal on Numerical Analysis*, 51(5):2887–2910, 2013.

[12] Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Applied Numerical Mathematics*, 25(2-3):151–167, 1997.

[13] Jianfang Lu, Yong Liu, and Chi-Wang Shu. An oscillation-free discontinuous Galerkin method for scalar hyperbolic conservation laws. *SIAM Journal on Numerical Analysis*, 59(3):1299–1324, 2021.

[14] Qiang Du, Lili Ju, and Jianfang Lu. A discontinuous Galerkin method for one-dimensional time-dependent nonlocal diffusion problems. *Mathematics of Computation*, 88(315):123–147, 2019.

[15] B. Sanderse, R. W. C. P. Verstappen, and B. Koren. Boundary treatment for fourth-order staggered mesh discretizations of the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 257(2):1472–1505, 2014.

[16] Alexander Naumann, Oliver Kolb, and Matteo Semplice. On a third order CWENO boundary treatment with application to networks of hyperbolic conservation laws. *Applied Mathematics and Computation*, 325:252–270, 2018.

[17] Haijin Wang, Qiang Zhang, and Chi-Wang Shu. Third order implicit–explicit Runge–Kutta local discontinuous Galerkin methods with suitable boundary treatment for

convection–diffusion problems with Dirichlet boundary conditions. *Journal of Computational and Applied Mathematics*, 342:164–179, 2018.

[18] Gourabananda Pahar and Anirban Dhar. Robust boundary treatment for open-channel flows in divergence-free incompressible SPH. *Journal of Hydrology*, 546:464–475, 2017.

[19] Yann Moguen, Pascal Bruel, and Erik Dick. Semi-implicit characteristic-based boundary treatment for acoustics in low Mach number flows. *Journal of Computational Physics*, 255(24):339–361, 2013.

[20] Z. Chen, C. Shu, D. Tan, and C. Wu. On improvements of simplified and highly stable lattice Boltzmann method: formulations, boundary treatment, and stability analysis. *International Journal for Numerical Methods in Fluids*, 87:161–179, 2017.

[21] Tao Xiong, Mengping Zhang, Yong-Tao Zhang, and Chi-Wang Shu. Fast sweeping fifth order WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment. *Journal of Scientific Computing*, 45(1):514–536, 2010.

[22] Cheng Liu and Changhong Hu. An efficient immersed boundary treatment for complex moving object. *Journal of Computational Physics*, 274:654–680, 2014.

[23] Mark H. Carpenter, David Gottlieb, and Saul Abarbanel. Stable and accurate boundary treatments for compact, high-order finite-difference schemes. *Applied Numerical Mathematics*, 12(1-3):55–87, 1993.

[24] Mark H. Carpenter, David Gottlieb, Saul Abarbanel, and Wai-Sun Don. The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error. *SIAM Journal on Scientific Computing*, 16(6):1241–1252, 1995.

[25] Weifeng Zhao and Juntao Huang. Boundary treatment of implicit-explicit Runge-Kutta method for hyperbolic systems with source terms. *Journal of Computational Physics*, 423:109828, 2020.

[26] Weifeng Zhao, Juntao Huang, and Steven J. Ruuth. Boundary treatment of high order Runge-Kutta methods for hyperbolic conservation laws. *Journal of Computational Physics*, 421:109697, 2020.

[27] Jianfang Lu, Jinwei Fang, Sirui Tan, Chi-Wang Shu, and Mengping Zhang. Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations. *Journal of Computational Physics*, 317:276–300, 2016.

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

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

[30] Tingting Li, Chi-Wang Shu, and Mengping Zhang. Stability analysis of the inverse Lax–Wendroff boundary treatment for high order central difference schemes for diffusion equations. *Journal of Scientific Computing*, 70(2):576–607, 2017.

[31] Gautier Dakin, Bruno Després, and Stéphane Jaouen. Inverse Lax-Wendroff boundary treatment for compressible Lagrange-remap hydrodynamics on Cartesian grids. *Journal of Computational Physics*, 353:228–257, 2018.

[32] Yupeng Ren, Tao Xiong, and Jianxian Qiu. A hybrid finite difference WENO-ZQ fast sweeping method for static Hamilton–Jacobi equations. *Journal of Scientific Computing*, 83:1–35, 2020.

[33] Shengrong Ding, Chi-Wang Shu, and Mengping Zhang. On the conservation of finite difference WENO schemes in non-rectangular domains using the inverse Lax-Wendroff boundary treatments. *Journal of Computational Physics*, 415:109516, 2020.

[34] Tingting Li, Jianfang Lu, and Chi-Wang Shu. Stability analysis of inverse Lax–Wendroff boundary treatment of high order compact difference schemes for parabolic equations. *Journal of Computational and Applied Mathematics*, 400:113711, 2022.

[35] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.

[36] X. Liu and Y. M. Zeng. Linear multistep methods for impulsive delay differential equations. *Applied Mathematics and Computation*, 321:555–563, 2018.

[37] Dong Wang and Steven J. Ruuth. Variable step-size implicit-explicit linear multistep methods for time-dependent partial differential equations. *Journal of Computational Mathematics*, 26(6):838–855, 2008.

[38] Ernst Hairer, Syvert Nørsett, and Gerhard Wanner. *Solving ordinary differential equations I: nonstiff problems*. Springer-Verlag, Berlin, 1993.

[39] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, 1984.