



Boundary treatment of implicit-explicit Runge-Kutta method for hyperbolic systems with source terms

Weifeng Zhao^a, Juntao Huang^{b,*}

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

^b Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA

ARTICLE INFO

Article history:

Received 25 April 2020

Received in revised form 6 August 2020

Accepted 2 September 2020

Available online 9 September 2020

Keywords:

Boundary treatment

Hyperbolic systems with source terms

Implicit-explicit Runge-Kutta method

High order finite difference

Inverse Lax-Wendroff procedure

ABSTRACT

In this paper, we develop a high order finite difference boundary treatment method for the implicit-explicit (IMEX) Runge-Kutta (RK) schemes solving hyperbolic systems with possibly stiff source terms on a Cartesian mesh. The main challenge is how to obtain the solutions at ghost points resulting from the wide stencil of the interior high order scheme. We address this problem by combining the idea of using the RK schemes at the boundary and an inverse Lax-Wendroff (ILW) procedure in [29,30]. Specifically, we only apply the ILW procedure in the starting stage of the RK method. In the intermediate stages, we solve out the intermediate solutions as well as their first-order spatial derivatives at the boundary by using the RK schemes, which are then used to compute solutions at ghost points by Taylor expansions. Our method is different from the widely used approach for the explicit RK schemes by imposing boundary conditions at intermediate stages [14,29,30] which does not apply to IMEX schemes. In addition, the intermediate boundary conditions are available for explicit RK schemes only up to third order while our method applies to IMEX and explicit RK schemes of arbitrary order. The stability and the desired accuracy of our boundary treatment with IMEX RK method up to fourth order are demonstrated numerically through 1D examples and 2D reactive Euler equations.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

Implicit-explicit (IMEX) Runge-Kutta (RK) schemes [7] are widely used in the time discretization of many partial differential equations (PDEs), such as convection-diffusion equations [15,25,32,33], hyperbolic systems with stiff sources [26,27,13,10,9,11,12], and the Boltzmann equation [16,19,18,23]. They achieve good stability and high order accuracy by treating the non-stiff terms explicitly and the stiff terms implicitly in the PDEs. Though powerful and efficient, the IMEX RK schemes for the time-dependent PDEs are not without drawbacks. A long-standing problem is the lack of appropriate boundary treatment method [7], which is fundamental for practical applications of IMEX RK schemes.

It is well-known that special care must be taken when imposing *time dependent* boundary conditions in the intermediate stages of explicit RK methods. The conventional and intuitive method would reduce the formal accuracy to first-order locally and second-order globally, independently of the spatial discretization [14]. A popular method to eliminate this problem is to impose consistent boundary conditions derived from the physical boundary condition and its derivatives for each intermediate stage [14]. However, this is a general remedy only for linear cases. For nonlinear hyperbolic systems, these

* Corresponding author.

E-mail addresses: wfzhao@ustb.edu.cn (W. Zhao), huangj75@msu.edu (J. Huang).

boundary conditions are only available for explicit RK schemes up to third order. The main reason is that beyond u_{tt} , one will need to use the “Jacobian of the Jacobian”, which cannot be related simply to the temporal derivatives of the vector u [14]. Though it is remedied for fourth order schemes in [1,28], the methods therein only apply to one-dimensional (1D) scalar equations or systems with all characteristics flowing into the domain at the boundary.

Another limitation of imposing consistent boundary conditions for intermediate stages [14] is that this strategy does not apply to IMEX RK schemes due to the fact that the implicit part would result in nonlinear equations which cannot be analytically solved in general. In [2–6], boundary conditions for intermediate stages are derived and analyzed for implicit RK schemes solving linear scalar equations. At this point, we remark that another limitation of all the boundary treatments mentioned above [14,1,28,2–6] is that they only deal with 1D case where the boundary is located at a grid point, restricting their applications for 2D problems on complex domains.

Perhaps due to the difficulties of extending the above boundary treatments of explicit and implicit schemes to the IMEX case, there are rare works on that of IMEX RK schemes. To the best of our knowledge, the only literature on this topic is [33]. In that reference, a strategy of imposing intermediate boundary conditions is presented for 1D linear convection-diffusion equations with discontinuous Galerkin (DG) method in spatial discretization. There are similar restrictions that it does not apply to general nonlinear equations and the physical boundary should be located at the grid point.

In this paper, we propose a boundary treatment method for the IMEX RK schemes solving hyperbolic systems with possibly stiff source terms. High order finite difference schemes on a Cartesian mesh are used for the space discretization. Due to the wide stencil of high order schemes, ghost points are needed near the boundary where the numerical stencil is partially outside of the computational domain. Then the main task of boundary treatment is to obtain the solutions at the ghost points. We address this problem by combining the idea of the inverse Lax-Wendroff (ILW) procedure [29,30] and using the semi-discrete schemes at the boundary. The ILW procedure is proposed in [29,30] as a technique of boundary treatment for hyperbolic systems. The idea is to apply the PDEs to convert spatial derivatives of inflow boundaries to time derivatives, while those of the outflow boundaries are extrapolation with the solutions at interior points. Nevertheless, the boundary treatment in [29,30] relies on the intermediate stage boundary conditions [14], which are not available for IMEX schemes as aforementioned. To overcome this difficulty, we propose to use the ILW procedure only in the starting stage of the RK scheme and solve out intermediate solutions as well as their first-order spatial derivatives by directly using the IMEX RK schemes at the boundary, instead of imposing intermediate-stage boundary conditions [14,29,30]. Then, we compute the solutions at ghost points by Taylor expansions at the boundary. In this way, our boundary treatment not only preserves the accuracy of the IMEX RK schemes but also possesses good stability. In addition, it allows the boundary to be arbitrarily located. While our method applies to general IMEX RK schemes, we demonstrate its effectiveness with three specific second-, third- and fourth-order schemes in [9,7,15]. The stability and the desired accuracy of the method are verified for both 1D and 2D cases, and the numerical results of the 2D reactive Euler equations are comparable to those obtained by the traditional reflective boundary conditions in [21].

We also point out the difference between our method and the boundary treatment in [33] if restricted to 1D case with grid point located at the boundary. The main idea in [33] is to replace the implicit part in the RK schemes using the PDEs and then represent the boundary values in the intermediate stages by physical boundary conditions and the derivatives. Instead of imposing intermediate-stage boundary conditions, we use the RK schemes at the boundary directly and solve out the solutions as well as their derivatives at the boundary in the intermediate stages. Moreover, the strategy in [33] assumes that the numerical solutions in the intermediate stages satisfy the PDEs. This does not hold true since the stage order is generally less than the order of accuracy of the RK method. This would cause the order reduction phenomenon for very high order schemes [36].

Since the IMEX RK schemes include explicit schemes as special cases, the present boundary treatment also applies to explicit RK schemes. It does not rely on intermediate-stage boundary conditions and thus is expected to be valid for explicit RK schemes of order higher than three. This is investigated in a separate work [36]. Therein we also show that the method in [29,30] suffers from order reduction in some cases while our method preserves the designed accuracy. Moreover, our method may be adapted to IMEX RK schemes solving the other partial differential equations, e.g. convection-diffusion equations [15,25,32,33] and the Boltzmann equation [16,19,18,23]. Many unresolved issues for the boundary treatment of RK schemes might be explored based on the idea of this work.

This paper is organized as follows. In Section 2, we introduce the IMEX RK schemes for hyperbolic systems with source terms. In Section 3, we use the 1D system to illustrate our idea of boundary treatment. In Section 4, we show the accuracy of our boundary treatment for a scalar hyperbolic equation with source term. The boundary treatment is then extended to the 2D reactive Euler equations in Section 5. Section 6 is devoted to numerical validations of our method. Finally, some conclusions and remarks are given in Section 7.

2. IMEX RK schemes for hyperbolic systems with source terms

Consider a one-dimensional nonlinear hyperbolic system with source term

$$\partial_t U + \partial_x F(U) = Q(U), \quad (2.1)$$

where $0 \leq x \leq 1$ and $U, F(U), Q(U) \in \mathbb{R}^M$. In the domain $0 \leq x \leq 1$, appropriate boundary conditions and initial data $U(0, x)$ are required to close the above system. We assume that the Jacobian matrix $F_U(U(t, 0))$ always has p positive eigenvalues and thus p independent relations among incoming and outgoing modes are given at the left boundary $x = 0$:

$$B(U(t, 0), t) = 0. \quad (2.2)$$

Similarly, the boundary conditions at $x = 1$ are also imposed properly.

A general semi-discrete IMEX RK scheme for the hyperbolic system (2.1) is [27]

$$U^{(i)} = U^n - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \partial_x F(U^{(j)}) + \Delta t \sum_{j=1}^i a_{ij} Q(U^{(j)}), \quad i = 1, 2, \dots, s, \quad (2.3a)$$

$$U^{n+1} = U^n - \Delta t \sum_{i=1}^s \tilde{\omega}_i \partial_x F(U^{(i)}) + \Delta t \sum_{i=1}^s \omega_i Q(U^{(i)}). \quad (2.3b)$$

Here the $s \times s$ matrices $\tilde{A} = (\tilde{a}_{ij})$ and $A = (a_{ij})$ satisfy $\tilde{a}_{ij} = 0$ for $j \geq i$ and $a_{ij} = 0$ for $j > i$ so that the scheme is explicit for the convection part and implicit for the source term. Along with the coefficient vectors $\tilde{w} = (\tilde{w}_1, \tilde{w}_2, \dots, \tilde{w}_s)^T$ and $w = (w_1, w_2, \dots, w_s)^T$, the scheme can be represented by a double Butcher tableau

$$\begin{array}{c|c} \tilde{c} & \tilde{A} \\ \hline & \tilde{w}^T \end{array} \quad \begin{array}{c|c} c & A \\ \hline & w^T \end{array}$$

where $\tilde{c} = (\tilde{c}_1, \tilde{c}_2, \dots, \tilde{c}_s)^T$, $c = (c_1, c_2, \dots, c_s)^T$ are defined as

$$\tilde{c}_i = \sum_{j=1}^{i-1} \tilde{a}_{ij}, \quad c_i = \sum_{j=1}^i a_{ij}. \quad (2.4)$$

The finite difference WENO scheme on a Cartesian mesh is applied for the space discretization [22], which needs a stencil of several points in each direction. As a consequence, ghost points are needed near the boundary $\partial\Omega$ where the numerical stencil is partially outside of Ω . For explicit RK schemes, the solutions at ghost points can be determined as in [29,30] with the aid of the ILW procedure and the boundary condition of each intermediate stage [14]. However, this approach does not apply to IMEX schemes since the boundary conditions of intermediate stages could not be analytically derived for general nonlinear systems. Thus boundary treatment of IMEX RK schemes remains a challenge and we address this problem in the following section.

3. Boundary treatment for 1D system

In this section, we use the one-dimensional system (2.1) to illustrate our idea of boundary treatment for the IMEX RK scheme (2.3), and then extend it to the two-dimensional case in Section 5. Without loss of generality, we employ the third-order finite difference WENO scheme in space [24] with $\Delta t = \mathcal{O}(\Delta x)$, while our method also applies to the fifth-order finite difference WENO scheme [22]. Both the third-order and the fifth-order WENO schemes will be verified numerically in Section 6.

3.1. Computation of solutions at ghost points

We focus on the left boundary $x = 0$ of the problem (2.1) and the method can be similarly applied to the right boundary. Following the notations in [30], we discretize the interval $(0, 1)$ by a uniform mesh

$$\frac{\Delta x}{2} = x_0 < x_1 < \dots < x_N = 1 - \frac{\Delta x}{2} \quad (3.1)$$

and set x_j , $j = -1, -2$ as two ghost points near the left boundary $x = 0$ (note that the boundary indeed can be arbitrarily located). Denote by U_j^n the numerical solution of U at position x_j and time t_n . Assume that the interior solutions U_j , $j = 0, 1, 2, \dots, N$ have been updated from time level t_{n-1} to time level t_n . As in [30], we use a third-order Taylor approximation to construct the values at the ghost points x_j for $j = -1, -2$, namely,

$$U_j^n = \sum_{k=0}^2 \frac{x_j^k}{k!} U^{n,(k)}, \quad j = -1, -2. \quad (3.2)$$

Here $U^{n,(k)}$ denotes a $(3-k)$ -th order approximation of the spatial derivative at the boundary point $\frac{\partial^k U}{\partial x^k} \big|_{x=0, t=t_n}$. With this formula, U_j^n at the ghost points can be obtained once $U^{n,(k)}$, $k = 0, 1, 2$ are provided, which is well treated by the ILW

procedure in [29,30] (see also the next subsection for details). Then we can obtain $U_j^{(1)}$ at the interior points by the RK scheme (2.3a). The next task is to compute $U_j^{(1)}$, $j = -1, -2$. We propose an approach to do this as follows.

Similar to the above procedure, we compute $U_j^{(1)}$ with the third-order Taylor expansion at the boundary point $x_b = 0$:

$$U_j^{(1)} = \sum_{k=0}^2 \frac{x_j^k}{k!} U^{(1),(k)}, \quad j = -1, -2, \quad (3.3)$$

where $U^{(1),(k)}$ denotes a $(3-k)$ -th order approximation of the spatial derivative $\frac{\partial^k U^{(1)}}{\partial x^k} \big|_{x=0}$. To compute $U^{(1),(k)}$ for $k = 0, 1, 2$, we apply the first stage of the semi-discrete scheme (2.3a) for $U^{(1)}$ at the boundary point $x_b = 0$:

$$U^{(1)}(x_b) = U^n(x_b) - \tilde{a}_{10} \Delta t \partial_x F(U^n(x_b)) + \Delta t a_{10} Q(U^{(1)}(x_b)). \quad (3.4)$$

Notice that $U^n(x_b) = U^{n,(0)} + \mathcal{O}(\Delta x^3)$ and $\partial_x F(U^n(x_b)) = F_U(U^n(x_b)) \partial_x U^n(x_b) = F_U(U^{n,(0)}) U^{n,(1)} + \mathcal{O}(\Delta x^2)$ are already known. Substituting these into (3.4) and solving the algebra equation, we can obtain an approximation of $U^{(1)}(x_b)$ and denote it by $U^{(1),(0)}$. In addition, the error between $U^{(1),(0)}$ and $U^{(1)}(x_b)$ defined by (3.4) is $\mathcal{O}(\Delta x^3)$.

Furthermore, taking derivatives with respect to x on both sides of (3.4) yields

$$\frac{\partial U^{(1)}}{\partial x} \big|_{x=x_b} = \partial_x U^n(x_b) - \tilde{a}_{10} \Delta t \partial_{xx} F(U^n(x_b)) + \Delta t a_{10} Q_U(U^{(1)}(x_b)) \frac{\partial U^{(1)}}{\partial x} \big|_{x=x_b}. \quad (3.5)$$

Here $\partial_x U^n(x_b) = U^{n,(1)} + \mathcal{O}(\Delta x^2)$ and $\partial_{xx} F(U^n(x_b)) = F_{UU}(U^n(x_b)) \partial_x U^n(x_b) \partial_x U^n(x_b) + F_U(U^n(x_b)) \partial_{xx} U^n(x_b) = F_{UU}(U^{n,(0)}) \times U^{n,(1)} U^{n,(1)} + F_U(U^{n,(0)}) U^{n,(2)} + \mathcal{O}(\Delta x)$. With these approximations, $\frac{\partial U^{(1)}}{\partial x} \big|_{x=x_b}$ can be solved out from (3.5) and the resulting solution $U^{(1),(1)}$ is a second-order approximation of $\frac{\partial U^{(1)}}{\partial x} \big|_{x=x_b}$.

By taking higher order derivatives on both sides of (3.4), one can also compute $\frac{\partial^k U^{(1)}}{\partial x^k} \big|_{x=x_b}$ for $k = 2$. However, this procedure is quite complicated as it involves Jacobian of Jacobian. Here we simply approximate $\frac{\partial^k U^{(1)}}{\partial x^k} \big|_{x=x_b}$ for $k = 2$ by using the $(3-k)$ -th order WENO type extrapolation with $U^{(1)}$ at interior points in [30], which will be introduced in subsection 3.3. In this way, we obtain $U^{(1),(k)}$ for $k = 0, 1, 2$ with accuracy of order $(3-k)$. Then $U_j^{(1)}$ for $j = -1, -2$ can be computed by (3.3) with third-order accuracy. Having $U^{(1)}$ at the ghost points, we can then evolve from $U^{(1)}$ to $U^{(2)}$ using the interior difference scheme.

Repeating the same procedure for each $U^{(i)}$, $2 \leq i \leq s$, we can compute the solution at the ghost points in the i -th intermediate stage. Then we can update the solution at all interior points in the $(i+1)$ -th stage. Finally, we obtain U^{n+1} , i.e., the solution at the end of a complete RK cycle.

We end this subsection with some remarks on the existence and the uniqueness of the solution for the algebra equations (3.4) and (3.5) as well as the implementation details.

Remark 3.1. It is shown in [34,35] that the stability of the hyperbolic system (2.1) requires that all the eigenvalues of the Jacobian matrix $Q_U(U)$ of the source term have nonpositive real parts. On the other hand, notice that the diagonal entries of the matrix A corresponding to the implicit part, i.e., a_{10} in (3.5), are usually nonnegative (see e.g. the three RK schemes in Section 6 below). Therefore, the matrix $(I - \Delta t a_{10} Q_U(U^{(1)}(x_b)))$ is invertible with any $\Delta t > 0$ and thus the equation (3.5) has a unique solution without any restriction on Δt .

Remark 3.2. The solvability of (3.4) is an issue at both boundary and interior points. When the source term of the hyperbolic system (2.1) linearly depends on U , then (3.4) has a unique solution by the same argument above. For nonlinear source terms, we recall from [34,35] that equations of the form (3.5) in many applications are endowed with equilibria U^{eq} such that $Q(U^{eq}) = 0$. When solutions of (3.5) are close to equilibria, the source term may be approximated by its linearization $Q(U) \approx Q(U^{eq}) + Q_U(U^{eq})(U - U^{eq}) = Q_U(U^{eq})(U - U^{eq})$. In this case, (3.4) has a unique solution as well. In general cases, the equation (3.4) has solutions when Δt is sufficiently small. Additionally, in our numerical experiments with CFL number being 0.8, the equation (3.4) can always be solved numerically and the good numerical results are generated. Namely, the existence of solutions to (3.4) does not impose additional restrictions on Δt from our numerical experience.

Remark 3.3. When solving (3.4), the MATLAB function *fsolve* is employed when the explicit expressions of solutions are unavailable. When using *fsolve* for (3.4), we set $U^n(x_b)$ as the initial guess and the correct solution of (3.4) is always obtained though it may have multiple solutions. Similarly, $U^{(i)}(x_b)$ is taken as the initial guess when solving $U^{(i+1)}(x_b)$ with *fsolve*.

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

For the sake of completeness, we provide the ILW procedure in [30] for computing $U^{n,(k)}$, $k = 0, 1, 2$, i.e., the $(3-k)$ -th order approximation of $\frac{\partial^k U^n}{\partial x^k}$ at the boundary $x = 0$.

3.2.1. $k = 0$

We first do a local characteristic decomposition to determine the inflow and outflow boundary conditions as in [30]. 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 $\lambda_1, \lambda_2, \dots, \lambda_p$ and $(M - p)$ negative eigenvalues $\lambda_{p+1}, \lambda_{p+2}, \dots, \lambda_M$ with l_1, l_2, \dots, l_p and $l_{p+1}, l_{p+2}, \dots, l_M$ the corresponding left eigenvectors, respectively. Define by $V_{j,m}$ the m -th component of local characteristic variable at grid point x_j , $j = 0, 1, 2$, i.e.,

$$V_{j,m} = l_m U_j^n, \quad m = 1, 2, \dots, M, \quad j = 0, 1, 2. \quad (3.6)$$

We extrapolate the outgoing characteristic variable $V_{j,m}$, $m = p + 1, p + 2, \dots, M$ to the boundary x_b with the WENO type extrapolation [30] (see also subsection 3.3), and denote the extrapolated k -th order derivative by

$$V_{x_b,m}^{*(k)}, \quad k = 0, 1, 2. \quad (3.7)$$

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

$$\begin{aligned} l_m U^{n,(0)} &= V_{x_b,m}^{*(0)}, \quad m = p + 1, p + 2, \dots, M, \\ B(U^{n,(0)}, t_n) &= 0. \end{aligned} \quad (3.8)$$

3.2.2. $k = 1$

Having $U^{n,(0)}$, we proceed to compute $U^{n,(1)}$ with the ILW procedure proposed in [29,30]. To do this, we take derivative with respect to t for the boundary condition (2.3)

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

which can be written as

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

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

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

With the above two equations and $V_{x_b,m}^{*(1)}$ obtained by the WENO type extrapolation, $U^{n,(1)}$ can be solved out from

$$l_m U^{n,(1)} = V_{x_b,m}^{*(1)}, \quad m = p + 1, p + 2, \dots, M, \quad (3.9a)$$

$$B_U(U^{n,(0)}, t_n) A(U^{n,(0)}) U^{n,(1)} = B_U(U^{n,(0)}, t_n) Q(U^{n,(0)}) - g(U^{n,(0)}, t_n). \quad (3.9b)$$

This is the ILW procedure in [29,30].

3.2.3. $k = 2$

Following [30], we approximate $\frac{\partial^2 U^n}{\partial x^2} \big|_{x=0}$ with the WENO type extrapolation. Specifically, since we have obtained the characteristic variables at grid points near the boundary in (3.6), the WENO type extrapolation is employed based on these characteristic variables to compute the second-order derivative $V_{x_b,m}^{*(2)}$ for each $m = 1, 2, \dots, M$. Then the approximation of $\frac{\partial^2 U^n}{\partial x^2} \big|_{x=0}$ is given by

$$U^{n,(2)} = L^{-1} V_{x_b,m}^{*(2)}, \quad (3.10)$$

where L is local left eigenvector matrix composed of l_1, l_2, \dots, l_M . Similarly, $\frac{\partial^k U^n}{\partial x^k} \big|_{x=0}$, $k > 2$ are also computed with the WENO type extrapolation if necessary for higher order boundary treatments.

3.3. WENO type extrapolation

Finally, we show how to obtain a $(3 - k)$ -th order approximation of $\frac{\partial^k V}{\partial x^k} \big|_{x=0}$, denoted by $V^{*(k)}$, $k = 0, 1, 2$, with the stencil $S_3 = \{x_0, x_1, x_2\}$, which is introduced in [29]. When V is smooth on S_3 , we compute $V^{*(k)}$ as

$$V^{*(k)} = \sum_{r=0}^2 d_r \frac{d^k p_r(x)}{dx^k} \big|_{x=0},$$

where $d_0 = \Delta x^2$, $d_1 = \Delta x$, $d_2 = 1 - \Delta x - \Delta x^2$ and $p_r(x)$ is the r -th order Lagrange polynomial constructed with S_3 . When there is a discontinuity in the stencil S_3 , the WENO type extrapolation is applied [29]:

$$V^{*(k)} = \sum_{r=0}^2 \omega_r \frac{d^k p_r(x)}{dx^k} \Big|_{x=0}, \quad (3.11)$$

where ω_r are nonlinear weights depending on the value of V_j . Similar to the classical WENO scheme, the nonlinear weights are given by [29]

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^2 \alpha_s}, \quad \alpha_r = \frac{d_r}{(\varepsilon + \beta_r)^2}.$$

Here $\varepsilon = 10^{-6}$ and β_r are the smoothness indicators determined by

$$\beta_0 = \Delta x^2, \quad \beta_r = \sum_{l=1}^r \int_{x_{-1}}^{x_0} \Delta x^{2l-1} \left(\frac{d^l}{dx^l} p_r(x) \right)^2 dx, \quad r = 1, 2.$$

The explicit expressions of β_1 and β_2 are given in [29]. When V is smooth on S_3 , it is shown in [29] that

$$\omega_0 = \mathcal{O}(\Delta x^2), \quad \omega_1 = \mathcal{O}(\Delta x), \quad \omega_2 = 1 - \omega_0 - \omega_1$$

so that the WENO type extrapolation (3.11) is $(3-k)$ -th order accurate.

We summarize our third-order boundary treatment for the hyperbolic system (2.1) as follows:

Step 1: Apply the ILW technique to compute $U^{n,(k)}$, i.e., the $(3-k)$ -th order approximation of $\frac{\partial^k U^n}{\partial x^k}$ at the boundary $x=0$ for $k=0, 1, 2$ as in subsection 3.2. Then impose U_j^n at the ghost points $j=-1, -2$ with the Taylor expansion (3.2). This is the same as that in [30] for explicit RK schemes. With U^n at the ghost points, we can evolve from U^n to $U^{(1)}$ with the interior difference scheme.

Step 2: By using the semi-discrete schemes in time (2.3) at the boundary point, compute $U^{(1),(k)}$, i.e., the $(3-k)$ -th order approximation of $\frac{\partial^k U^{(1)}}{\partial x^k}$ at the boundary $x=0$ for $k=0, 1, 2$ as in subsection 3.1. Specifically, $U^{(1),(0)}$ and $U^{(1),(1)}$ are solved out from the semi-discrete RK solver (3.4) and (3.5), respectively, and $U^{(1),(2)}$ is approximated with the WENO type extrapolation in subsection 3.3. Then impose $U_j^{(1)}$ at the ghost points $j=-1, -2$ with the Taylor expansion (3.3). Having $U^{(1)}$ at the ghost points, we can update $U^{(2)}$ at all interior points.

Step 3: Repeat the same procedure as in Step 2 for each $U^{(i)}$, $2 \leq i \leq s$ to compute the solution at the ghost points in the i -th intermediate stage. Then we can update the solution at all interior points in the $(i+1)$ -th stage. Finally, we obtain U^{n+1} , i.e., the solution at the end of a complete RK cycle.

Remark 3.4. Combining the idea of using the RK schemes of intermediate solutions at the boundary and the ILW procedure is the key of our boundary treatment. The former preserves the accuracy of the RK method and the latter guarantees the stability. This is different from the widely used approach for the explicit RK scheme by imposing boundary conditions for intermediate solutions [14], which are unavailable for the IMEX schemes. In addition, the intermediate boundary conditions are only available for explicit RK schemes up to third order while our method applies to general IMEX RK scheme.

Remark 3.5. Though the idea of using the RK schemes of intermediate solutions has been stressed in [28], the method therein is only valid for 1D scalar equation or 1D system with all characteristics flowing into the domain at the boundary. Another limitation of the method in [28] is that the boundary has to be located at a grid point, which makes it difficult to be extended to 2D case.

Remark 3.6. Since the IMEX RK schemes include explicit schemes as special cases, the present boundary treatment also applies to explicit RK schemes. It does not need to impose intermediate boundary conditions and thus is expected to be valid for explicit RK schemes of order higher than three. This is investigated in a separate work [36].

4. Accuracy analysis of the boundary treatment

In this section, we show the accuracy of our boundary treatment for the scalar hyperbolic equation with source term

$$u_t + f(u)_x = q(u), \quad x \in (0, 1), \quad t > 0. \quad (4.1)$$

We assume that the left boundary $x=0$ is an inflow boundary and the right one is an outflow boundary, and we only consider the boundary treatment on the left side. The boundary condition at $x=0$ is given by

$$u(t, 0) = g(t). \quad (4.2)$$

Here we use the third-order IMEX-RK scheme in [7] (the corresponding double Butcher tableau is given in Section 6) and take $\Delta t = \mathcal{O}(\Delta x)$. It should be noted that this illustration could be carried out for any IMEX RK schemes.

We first discretize (4.1) in time:

$$W^{(1)} = W^n, \quad (4.3a)$$

$$W^{(2)} = W^n - \Delta t \frac{1}{2} f(W^{(1)})_x + \Delta t \frac{1}{2} q(W^{(2)}), \quad (4.3b)$$

$$W^{(3)} = W^n - \Delta t \left(\frac{11}{18} f(W^{(1)})_x + \frac{1}{18} f(W^{(2)})_x \right) + \Delta t \left(\frac{1}{6} q(W^{(2)}) + \frac{1}{2} q(W^{(3)}) \right), \quad (4.3c)$$

$$W^{(4)} = W^n - \Delta t \left(\frac{5}{6} f(W^{(1)})_x - \frac{5}{6} f(W^{(2)})_x + \frac{1}{2} f(W^{(3)})_x \right) + \Delta t \left(-\frac{1}{2} q(W^{(2)}) + \frac{1}{2} q(W^{(3)}) + \frac{1}{2} q(W^{(4)}) \right), \quad (4.3d)$$

$$W^{(5)} = W^n - \Delta t \left(\frac{1}{4} f(W^{(1)})_x + \frac{7}{4} f(W^{(2)})_x + \frac{3}{4} f(W^{(3)})_x - \frac{7}{4} f(W^{(4)})_x \right) + \Delta t \left(\frac{3}{2} q(W^{(2)}) - \frac{3}{2} q(W^{(3)}) + \frac{1}{2} q(W^{(4)}) + \frac{1}{2} q(W^{(5)}) \right), \quad (4.3e)$$

$$W^{n+1} = W^n - \Delta t \left(\frac{1}{4} f(W^{(1)})_x + \frac{7}{4} f(W^{(2)})_x + \frac{3}{4} f(W^{(3)})_x - \frac{7}{4} f(W^{(4)})_x \right) + \Delta t \left(\frac{3}{2} q(W^{(2)}) - \frac{3}{2} q(W^{(3)}) + \frac{1}{2} q(W^{(4)}) + \frac{1}{2} q(W^{(5)}) \right). \quad (4.3f)$$

Here $W^n (= W^{(1)}) = W^n(x)$ is the solution at time t_n and $W^{(i)} = W^{(i)}(x)$ for $i = 2, 3, 4, 5$ are intermediate solutions, which are functions defined in the domain $(0, 1)$. Note that $W^n (= W^{(1)})$ is supplied with boundary condition $W^n(0) = g(t_n)$ while no boundary conditions are needed for intermediate solutions $W^{(i)} = W^{(i)}(x)$, $i = 2, 3, 4, 5$. Once $W^n (= W^{(1)})$ is given, $W^{(2)}$ is determined by (4.3b) and so on for $W^{(i)}$ for $i = 3, 4, 5$ and W^{n+1} . The local truncation error of (4.3) can be easily verified to be $\mathcal{O}(\Delta t^4)$ by plugging $W^n(x) = u(t_n, x)$ into (4.3), due to the third-order accuracy of the IMEX RK scheme [7].

Next, we perform spatial discretization on (4.3) with the mesh given by (3.1) and have the fully discrete scheme:

$$U_j^{(1)} = U_j^n, \quad (4.4a)$$

$$U_j^{(2)} = U_j^n - \Delta t \frac{1}{2} D_j^{(1)} + \Delta t \frac{1}{2} q(U_j^{(2)}), \quad (4.4b)$$

$$U_j^{(3)} = U_j^n - \Delta t \left(\frac{11}{18} D_j^{(1)} + \frac{1}{18} D_j^{(2)} \right) + \Delta t \left(\frac{1}{6} q(U_j^{(2)}) + \frac{1}{2} q(U_j^{(3)}) \right), \quad (4.4c)$$

$$U_j^{(4)} = U_j^n - \Delta t \left(\frac{5}{6} D_j^{(1)} - \frac{5}{6} D_j^{(2)} + \frac{1}{2} D_j^{(3)} \right) + \Delta t \left(-\frac{1}{2} q(U_j^{(2)}) + \frac{1}{2} q(U_j^{(3)}) + \frac{1}{2} q(U_j^{(4)}) \right), \quad (4.4d)$$

$$U_j^{(5)} = U_j^n - \Delta t \left(\frac{1}{4} D_j^{(1)} + \frac{7}{4} D_j^{(2)} + \frac{3}{4} D_j^{(3)} - \frac{7}{4} D_j^{(4)} \right) + \Delta t \left(\frac{3}{2} q(U_j^{(2)}) - \frac{3}{2} q(U_j^{(3)}) + \frac{1}{2} q(U_j^{(4)}) + \frac{1}{2} q(U_j^{(5)}) \right), \quad (4.4e)$$

$$U_j^{n+1} = U_j^n - \Delta t \left(\frac{1}{4} D_j^{(1)} + \frac{7}{4} D_j^{(2)} + \frac{3}{4} D_j^{(3)} - \frac{7}{4} D_j^{(4)} \right) + \Delta t \left(\frac{3}{2} q(U_j^{(2)}) - \frac{3}{2} q(U_j^{(3)}) + \frac{1}{2} q(U_j^{(4)}) + \frac{1}{2} q(U_j^{(5)}) \right) \quad (4.4f)$$

for $j = 0, 1, 2, \dots, N$. Here U_j^n is the numerical solution at time t_n and position x_j , $U_j^{(i)}$ is the approximation of $W^{(i)}$ at x_j , and $D_j^{(i)} := D_j^{(i)}(U_j^{(i-2)}, U_j^{(i-1)}, U_j^{(i)}, U_j^{(i+1)}, U_j^{(i+2)})$ is the third-order WENO scheme that approximates $\partial_x f(W^{(i)})$ at x_j and satisfies $D_j^{(i)}(W^{(i)}(x_{j-2}), W^{(i)}(x_{j-1}), W^{(i)}(x_j), W^{(i)}(x_{j+1}), W^{(i)}(x_{j+2})) = \partial_x f(W^{(i)}(x_j)) + \mathcal{O}(\Delta x^3)$. Note that $D_0^{(i)}$ and $D_1^{(i)}$ need solutions at ghost points $U_{-2}^{(i)}$ and $U_{-1}^{(i)}$, which are specified by our boundary treatment in the previous section.

Since the semi-discrete scheme (4.3) is a third-order approximation of the scalar equation (4.1) with the boundary condition (4.2), we only need analyze the difference between (4.3) and (4.4) together with our boundary treatment. To do

this, we plug the exact solutions $U_j^n (= U_j^{(1)}) = u(t_n, x_j)$, $U_j^{n+1} = u(t_{n+1}, x_j)$ and $U_j^{(i)} = W^{(i)}(x_j)$, $i = 2, 3, 4, 5$ (which are the solutions of (4.3b)–(4.3e) under $W^n(x) = u(t_n, x)$) into (4.4) and investigate the residuals of each stage.

Let us first consider the second stage (4.4b). For $j \geq 2$, $D_j^{(1)}$ does not involve ghost points and the residual is $\mathcal{O}(\Delta t \Delta x^3) = \mathcal{O}(\Delta x^4)$. Then we just need to investigate the residuals of (4.4) for $j = 0$ and 1. According to our boundary treatment in Section 3, $U_j^n (= U_j^{(1)})$ for $j = -1, -2$ are computed with the third-order Taylor expansion (3.2) at the boundary $x_b = 0$, where the value at $x_b = 0$ is given by the boundary condition (4.2), the first-order derivative is obtained by the ILW procedure which only uses the equation (4.1) and the boundary condition (4.2), and the second-order derivative is obtained with the extrapolation of U_j^n for $j = 0, 1, 2$. Namely, U_j^n for $j = -1, -2$ depend on the boundary value $g(t_n)$, its derivative $g'(t_n)$ and the interior solutions U_j^n for $j = 0, 1, 2$. For convenience, we denote the dependence by two functions $U_{-1}^n = T_{-1}(g(t_n), g'(t_n), U_0^n, U_1^n, U_2^n)$ and $U_{-2}^n = T_{-2}(g(t_n), g'(t_n), U_0^n, U_1^n, U_2^n)$, and let $T_{-1} = T_{-1}(g(t_n), g'(t_n), W^{(1)}(x_0), W^{(1)}(x_1), W^{(1)}(x_2))$ and $T_{-2} = T_{-2}(g(t_n), g'(t_n), W^{(1)}(x_0), W^{(1)}(x_1), W^{(1)}(x_2))$. Since we use the third-order Taylor expansion, the difference between $D_j^{(1)}(T_{-2}, T_{-1}, W^{(1)}(x_0), W^{(1)}(x_1), W^{(1)}(x_2))$ and $\partial_x f(W^{(1)}(x_0))$ is $\mathcal{O}(\Delta x^2)$. Thus, the residual of (4.4b) is $\mathcal{O}(\Delta t \Delta x^2) = \mathcal{O}(\Delta x^3)$ for $j = 0$. One can similarly show that the residual of (4.4b) for $j = 1$ is also $\mathcal{O}(\Delta x^3)$ as it involves the ghost point solution $U_{-1}^{(1)}$.

Next, we move to the third stage (4.4c). In this stage, the ghost point solutions $U_j^{(2)}$ for $j = -1, -2$ are also computed by a third-order Taylor expansion at the boundary $x_b = 0$. Here the value of $U^{(2)}$ at the boundary, denoted by $U_b^{(2)}$, is obtained by the RK scheme (4.3b) at the boundary. The first-order derivative $U_b^{(2),(1)}$ is computed by taking derivative on both sides of (4.3b) and its error is $\mathcal{O}(\Delta x^3)$ (see (3.5) and the explanations just below). The second-order derivative $U_b^{(2),(2)}$ is obtained with the first-order extrapolation. Then $U_j^{(2)}$ for $j = -1, -2$ depend on $g(t_n)$, $g'(t_n)$ and interior solutions $U_j^{(1)}$, $U_j^{(2)}$ for $j = 0, 1, 2$.¹ We denote the dependence by two functions $U_{-1}^{(2)} = G_{-1}(g(t_n), g'(t_n), U_0^{(1)}, U_1^{(1)}, U_2^{(1)}, U_0^{(2)}, U_1^{(2)}, U_2^{(2)})$ and $U_{-2}^{(2)} = G_{-2}(g(t_n), g'(t_n), U_0^{(1)}, U_1^{(1)}, U_2^{(1)}, U_0^{(2)}, U_1^{(2)}, U_2^{(2)})$, and set

$$\begin{aligned} G_{-1} &= G_{-1}(g(t_n), g'(t_n), W^{(1)}(x_0), W^{(1)}(x_1), W^{(1)}(x_2), W^{(2)}(x_0), W^{(2)}(x_1), W^{(2)}(x_2)), \\ G_{-2} &= G_{-2}(g(t_n), g'(t_n), W^{(1)}(x_0), W^{(1)}(x_1), W^{(1)}(x_2), W^{(2)}(x_0), W^{(2)}(x_1), W^{(2)}(x_2)). \end{aligned}$$

Then the third-order Taylor expansion yields that the difference between $D_j^{(2)}(G_{-2}, G_{-1}, W^{(2)}(x_0), W^{(2)}(x_1), W^{(2)}(x_2))$ and $\partial_x f(W^{(2)}(x_0))$ is $\mathcal{O}(\Delta x^2)$. Thus, the residual of (4.4c) is $\mathcal{O}(\Delta t \Delta x^2) = \mathcal{O}(\Delta x^3)$ for $j = 0$. Similarly, the residual of (4.4b) for $j = 1$ is also $\mathcal{O}(\Delta x^3)$ as it involves the ghost point solution $U_{-2}^{(1)}$. For grid points $j \geq 2$, the residual is $\mathcal{O}(\Delta x^4)$.

Using the same argument, one can see that the residuals of the remaining intermediate stages (4.4d) and (4.4e) are $\mathcal{O}(\Delta x^3)$ for $j = 0, 1$ and $\mathcal{O}(\Delta x^4)$ for $j \geq 2$. Finally, absorbing the time discretization error, the residual of the last step (4.4f) is $\mathcal{O}(\Delta t^3)$ (or $\mathcal{O}(\Delta x^3)$) for $j = 0, 1$ and $\mathcal{O}(\Delta t^4)$ (or $\mathcal{O}(\Delta x^4)$) for $j \geq 2$, under the relation $\Delta t = \mathcal{O}(\Delta x)$. Notice that the residual of $\mathcal{O}(\Delta t^3)$ is local (near the boundary) and thus the overall residual is $\mathcal{O}(\Delta t^4)$ [17]. This indicates the third-order accuracy of our boundary treatment.

Remark 4.1. Since the IMEX RK schemes include the explicit RK schemes as special cases, the present boundary treatment also applies to explicit schemes. This is investigated in our separate work [36]. In [36], we show that there exists a difference of order $\mathcal{O}(\Delta t^2)$ for $u_x^{(1)}$, i.e., the first-order derivative of the solution in the first stage, between our method and the ILW method in [29,30], when applying to the third-order strong-stability-preserving (SSP) RK scheme solving 1D scalar conservation laws. This would eventually result a difference of $\mathcal{O}(\Delta t^2 \Delta x)$ in $u^{(1)}$ at ghost points computed with Taylor expansion. The order reduction phenomenon will arise for the ILW method in [29,30] in some cases, while our method preserves the full order of accuracy. We refer readers to [36] for more details.

5. Boundary treatment for 2D reactive Euler equations

In this section, we extend the above idea of boundary treatment to the 2D case. For convenience of representation, we concentrate on the 2D reactive Euler equations

$$\partial_t U + \partial_x F(U) + \partial_y G(U) = S(U), \quad (5.1)$$

where

¹ Here $g(t_n)$ and $g'(t_n)$ are for the computation of $U_b^{(2)}$; $U_j^{(1)}$, $j = 0, 1, 2$ are used to compute the second-order derivative of $U^{(1)}$ at the boundary, which is needed for $U_b^{(2),(1)}$; and $U_j^{(2)}$, $j = 0, 1, 2$ are used to obtain $U_b^{(2),(2)}$.

$$\begin{aligned}
U &= (\rho, \rho u, \rho v, E, \rho Y)^T, \\
F(U) &= (\rho u, \rho u^2 + p, \rho uv, (E + p)u, \rho u Y)^T, \\
G(U) &= (\rho v, \rho uv, \rho v^2 + p, (E + p)v, \rho v Y)^T, \\
S(U) &= (0, 0, 0, 0, \omega)
\end{aligned}$$

and $E = \frac{1}{2}\rho(u^2 + v^2) + \frac{p}{\gamma-1} + \rho q Y$. 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, Y is the reactant mass fraction, $q > 0$ is the heat release of reaction and γ is the specific heat ratio. The source term is assumed to be in an Arrhenius form

$$\omega = -\tilde{K}\rho Y e^{-\tilde{T}/T},$$

where $T = p/\rho$ is the temperature, $\tilde{T} > 0$ is the activation constant temperature and $\tilde{K} > 0$ is a constant rate coefficient.

The IMEX RK scheme for the 2D reactive Euler equations (5.1) is

$$U^{(i)} = U^n - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \partial_x F(U^{(j)}) - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \partial_y G(U^{(j)}) + \Delta t \sum_{j=1}^i a_{ij} S(U^{(j)}), \quad i = 1, 2, \dots, s, \quad (5.2a)$$

$$U^{n+1} = U^n - \Delta t \sum_{i=1}^s \tilde{\omega}_i \partial_x F(U^{(i)}) - \Delta t \sum_{i=1}^s \tilde{\omega}_i \partial_y G(U^{(i)}) + \Delta t \sum_{i=1}^s \omega_i S(U^{(i)}), \quad (5.2b)$$

where the coefficients \tilde{a}_{ij} , a_{ij} , $\tilde{\omega}_i$ and ω_i are the same with those in (2.3). Assume the solutions of all the grid points inside Ω have been updated from time level t_{n-1} to t_n . For a ghost point P , we find a point $P_0 = (x_0, y_0) = \mathbf{x}_0$ on the boundary $\partial\Omega$ such that the normal $\mathbf{n}(\mathbf{x}_0)$ at P_0 goes through P . Here $\mathbf{n}(\mathbf{x}_0)$ points to the exterior of Ω . We set up a local coordinate system at P_0 by

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \end{pmatrix}, \quad (5.3)$$

where θ is the angle between the normal $\mathbf{n}(\mathbf{x}_0)$ and the x -axis, and \mathbf{T} is a rotational matrix. The \hat{x} -axis then points in the same direction as $\mathbf{n}(\mathbf{x}_0)$ and the \hat{y} -axis points in the tangent direction. In this local coordinate system, the reactive Euler equations (5.1) become

$$\hat{U}_t + F(\hat{U})_{\hat{x}} + G(\hat{U})_{\hat{y}} = S(\hat{U}) \quad (5.4)$$

with

$$\hat{U} = (\rho, \rho \hat{u}, \rho \hat{v}, E, \rho Y)^T, \quad \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} = \mathbf{T} \begin{pmatrix} u \\ v \end{pmatrix}.$$

The IMEX scheme in this coordinate is

$$\hat{U}^{(i)} = \hat{U}^n - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \partial_{\hat{x}} F(\hat{U}^{(j)}) - \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} \partial_{\hat{y}} G(\hat{U}^{(j)}) + \Delta t \sum_{j=1}^i a_{ij} S(\hat{U}^{(j)}), \quad i = 1, 2, \dots, s, \quad (5.5a)$$

$$\hat{U}^{n+1} = \hat{U}^n - \Delta t \sum_{i=1}^s \tilde{\omega}_i \partial_{\hat{x}} F(\hat{U}^{(i)}) - \Delta t \sum_{i=1}^s \tilde{\omega}_i \partial_{\hat{y}} G(\hat{U}^{(i)}) + \Delta t \sum_{i=1}^s \omega_i S(\hat{U}^{(i)}). \quad (5.5b)$$

For a third-order boundary treatment, \hat{U}^n at the ghost point P is imposed by the Taylor expansion

$$\hat{U}^n(P) = \sum_{k=0}^2 \frac{\Delta^k}{k!} \hat{U}^{n,(k)}, \quad (5.6)$$

where Δ is the \hat{x} -coordinate of P and $\hat{U}^{n,(k)}$ is a $(3-k)$ -th order approximation of the normal derivative $\frac{\partial^k \hat{U}}{\partial \hat{x}^k} \Big|_{(x,y)=\mathbf{x}_0, t=t_n}$. With \hat{U}^n at ghost points, $\hat{U}^{(1)}$ at interior points can be obtained with the scheme (5.5a). Since the computation of $\hat{U}^{n,(k)}$ is the same with that in [30] and its basic idea has been illustrated for the 1D case, we omit the details here.

The next step is to compute $\hat{U}^{(1)}$ at the ghost points. Similar to the 1D case, we use the Taylor expansion

$$\hat{U}^{(1)}(P) = \sum_{k=0}^2 \frac{\Delta^k}{k!} \hat{U}^{(1),(k)} \quad (5.7)$$

with $\hat{U}^{(1),(k)}$ a $(3-k)$ -th order approximation of $\frac{\partial^k \hat{U}^{(1)}}{\partial \hat{x}^k} \big|_{(x,y)=\mathbf{x}_0}$, and turn to determine $\hat{U}^{(1),(k)}$, $k = 0, 1, 2$. First, $\hat{U}^{(1),(0)} = \hat{U}^{(1)}(P_0)$ can be computed with scheme (5.5a), i.e.,

$$\hat{U}^{(1)}(P_0) = \hat{U}^n(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{x}} F(\hat{U}^n)(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{y}} G(\hat{U}^n)(P_0) + \Delta t a_{11} S(\hat{U}^{(1)})(P_0). \quad (5.8)$$

Furthermore, taking derivatives with respect to \hat{x} and \hat{y} on both sides of the above equation, we obtain

$$\begin{aligned} \partial_{\hat{x}} \hat{U}^{(1)}(P_0) &= \partial_{\hat{x}} \hat{U}^n(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{x}\hat{x}} F(\hat{U}^n)(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{x}\hat{y}} G(\hat{U}^n)(P_0) + \Delta t a_{11} S_U(\hat{U}^{(1)}) \partial_{\hat{x}} \hat{U}^{(1)}(P_0), \\ \partial_{\hat{y}} \hat{U}^{(1)}(P_0) &= \partial_{\hat{y}} \hat{U}^n(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{x}\hat{y}} F(\hat{U}^n)(P_0) - \Delta t \tilde{a}_{11} \partial_{\hat{y}\hat{y}} G(\hat{U}^n)(P_0) + \Delta t a_{11} S_U(\hat{U}^{(1)}) \partial_{\hat{y}} \hat{U}^{(1)}(P_0). \end{aligned} \quad (5.9)$$

Since $\hat{U}^{(1)}(P_0)$ has already been obtained, and $\partial_{\hat{x}\hat{x}} F(\hat{U}^n)(P_0)$, $\partial_{\hat{x}\hat{y}} G(\hat{U}^n)(P_0)$, $\partial_{\hat{x}\hat{y}} F(\hat{U}^n)(P_0)$ and $\partial_{\hat{y}\hat{y}} G(\hat{U}^n)(P_0)$ can be approximated with \hat{U}^n at interior grid points by the 2D WENO type extrapolation [30], then $\partial_{\hat{x}} \hat{U}^{(1)}(P_0)$ and $\partial_{\hat{y}} \hat{U}^{(1)}(P_0)$ can be solved out from the above equation. High order derivatives, $\partial_{\hat{x}\hat{x}} \hat{U}^{(1)}(P_0)$, $\partial_{\hat{x}\hat{y}} \hat{U}^{(1)}(P_0)$ and $\partial_{\hat{y}\hat{y}} \hat{U}^{(1)}(P_0)$, are also computed by the 2D WENO type extrapolation [30]. Then $\hat{U}^{(1)}(P)$ can be computed with (5.7), and similarly $\hat{U}^{(k)}(P)$, $k \geq 2$ can be obtained. The velocities at the ghost point can be transformed back to the original coordinate by

$$\begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{T}^{-1} \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix}$$

while the scalar quantities remain unchanged in the two coordinates. In this way, both the solutions U^n at time t_n and intermediate solutions $U^{(i)}$ at ghost points are determined. Note that the 2D WENO type extrapolation has been introduced detailedly in [30] and thus is omitted here.

6. Numerical validations

In this section we conduct several numerical experiments for both 1D and 2D problems to demonstrate the numerical stability and accuracy of our boundary treatment method. Specifically, we test our method with three IMEX schemes: the second-order IMEX RK scheme in [9] with the double Butcher tableau

0	0	0	0	0	0	0	0	0	0	0	0
1/4	1/4	0	0	0	0	1/4	0	1/4	0	0	0
1/3	1/6	1/6	0	0	0	1/3	0	1/12	1/4	0	0
2/3	-2/3	0	4/3	0	0	2/3	0	-11/12	4/3	1/4	0
1	-1/16	1/2	0	9/16	0	1	0	9/31	12/31	9/124	1/4
	-1/16	1/2	0	9/16	0		0	9/31	12/31	9/124	1/4

the third-order IMEX RK scheme in [7] with the double Butcher tableau

0	0	0	0	0	0	0	0	0	0	0	0
1/2	1/2	0	0	0	0	1/2	0	1/2	0	0	0
2/3	11/18	1/18	0	0	0	2/3	0	1/6	1/2	0	0
1/2	5/6	-5/6	1/2	0	0	1/2	0	-1/2	1/2	1/2	0
1	1/4	7/4	3/4	-7/4	0	1	0	3/2	-3/2	1/2	1/2
	1/4	7/4	3/4	-7/4	0		0	3/2	-3/2	1/2	1/2

and the fourth-order IMEX RK scheme in [15] with the double Butcher tableau

0	0	0	0	0	0	0	0	0	0	0	0
1/2	1/2	0	0	0	0	1/2	0	1/2	0	0	0
83/250	13861/62500	6889/62500	0	0	0	83/250	0	0	0	0	0
31/50	-116923316275/2393684061468	-2731218467317/15368042101831	9408046702089/11113171139209	0	0	31/50	0	0	0	0	0
17/20	-451086348788/2902428689909	-2682348792572/7519795681897	12662868775082/11960479115383	3355817975965/11060851509271	0	17/20	0	0	0	0	0
1	647845179188/3216320057751	73281519250/8382639484533	552539513391/3454668386233	3354512671639/8306763924573	4040/17871	1	0	0	0	0	0
	82889/524892	0	15625/83664	69875/102672	-2260/8211		82889/524892	0	15625/83664	69875/102672	-2260/8211

0	0	0	0	0	0	0
$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0
$\frac{83}{250}$	$\frac{8611}{62500}$	$-\frac{1743}{31250}$	$\frac{1}{4}$	0	0	0
$\frac{31}{50}$	$\frac{5012029}{34652500}$	$-\frac{654441}{2922500}$	$\frac{174375}{388108}$	$\frac{1}{4}$	0	0
$\frac{17}{20}$	$\frac{15267082809}{155376265600}$	$-\frac{71443401}{120774400}$	$\frac{730878875}{902184768}$	$\frac{2285395}{8070912}$	$\frac{1}{4}$	0
1	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$
	$\frac{82889}{524892}$	0	$\frac{15625}{83664}$	$\frac{69875}{102672}$	$-\frac{2260}{8211}$	$\frac{1}{4}$

It is clear that the second- and third-order schemes satisfy $\tilde{a}_{si} = w_i, a_{si} = w_i, i = 1, 2, \dots, s$, i.e., $U^{n+1} = U^{(s)}$. Such schemes are said to be globally stiffly accurate (GSA) as they preserve the order of accuracy when the source term is highly stiff [8,9]. The first two schemes are denoted by GSA(4,4,2) and GSA(4,4,3), respectively, and the fourth-order scheme is denoted by ARK(6,6,4) (ARK: Additive Runge-Kutta; (s, r, q) : s is the number of stages in the explicit scheme, r is the number of stages in the implicit scheme, and q is the order of the IMEX scheme). Notice that the ARK(6,6,4) scheme is not globally stiffly accurate since $U^{n+1} \neq U^{(s)}$ [8,9].

For the spatial discretization, we use the third-order finite difference WENO scheme [24] for GSA(4,4,2) and GSA(4,4,3), and the fifth-order WENO scheme [22] for ARK(6,6,4). The Lax-Friedrichs flux splitting is employed to form the numerical fluxes [22]. Moreover, we fix the CFL number to be 0.8 in all the numerical examples. For 1D examples, the mesh for computation is taken as (3.1), namely, the boundary is located in the middle of two grid points. Different locations of the boundary are investigated for 2D problems.

6.1. 1D examples

Example 1. We first consider a 1D scalar hyperbolic equation

$$\partial_t u + u \partial_x u = u^2 + u, \quad (6.1)$$

in the domain $0 \leq x \leq 1$ with the exact solution $u = \exp(t + x)$. Since $u > 0$, the boundary condition only needs to be imposed at $x = 0$. Specifically, the initial and boundary conditions are set as

$$u(x, 0) = \exp(x), \quad u(0, t) = \exp(t) \quad (6.2)$$

to match the exact solution. We compute the L^1 , L^2 and L^∞ errors at the terminal time $t = 1$, and the results are listed in Table 6.1. It can be seen that the convergence order is around three for the GSA(4,4,3) scheme. For the GSA(4,4,2) scheme, the convergence order is about three for large Δx and close to two when Δx is small, which may be due to the fact that the spatial error dominates the temporal one. Also for this reason, the convergence order for the ARK(6,6,4) scheme is about five.

Table 6.1
Example 1: Error table for the scalar nonlinear equation (6.1).

	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	8.22e-4		1.36e-3		4.36e-3	
	1/40	1.29e-4	2.67	2.35e-4	2.53	1.00e-3	2.12
	1/80	1.82e-5	2.83	3.09e-5	2.93	1.19e-4	3.07
	1/160	2.53e-6	2.84	3.78e-6	3.03	1.02e-5	3.55
	1/320	4.19e-7	2.60	5.83e-7	2.70	1.30e-6	2.97
	1/640	8.33e-7	2.33	1.12e-7	2.38	2.64e-7	2.30
GSA(4,4,3)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	7.62e-4		1.29e-3		4.22e-3	
	1/40	1.13e-4	2.75	2.17e-4	2.57	9.57e-4	2.14
	1/80	1.41e-5	3.01	2.59e-5	3.07	1.07e-4	3.16
	1/160	1.50e-6	3.23	2.42e-6	3.42	7.20e-6	3.89
	1/320	1.57e-7	3.25	2.32e-7	3.38	5.53e-7	3.70
ARK(6,6,4)	1/640	1.74e-8	3.18	2.41e-8	3.27	5.44e-8	3.34
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	3.27e-6		5.13e-6		1.65e-5	
	1/40	1.24e-7	4.72	2.18e-7	4.56	8.90e-7	4.21
	1/80	4.40e-9	4.82	8.80e-9	4.63	4.52e-8	4.30
	1/160	1.25e-10	5.13	2.59e-10	5.09	1.25e-9	5.18
	1/320	2.78e-12	5.50	5.24e-12	5.63	2.14e-11	5.87

Table 6.2

Example 1: Error table for the scalar nonlinear equation (6.1) with the source term replaced by $-0.1 \exp(u) + \sin(u)$.

	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	2.59e-4		5.93e-4		2.01e-3	
	1/40	1.85e-5	3.80	3.50e-5	4.08	1.73e-4	3.54
	1/80	2.43e-6	2.93	2.85e-6	3.62	6.71e-6	4.69
	1/160	4.40e-7	2.46	5.02e-7	2.50	9.73e-7	2.79
GSA(4,4,3)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	1.75e-4		4.01e-4		1.43e-3	
	1/40	7.42e-6	4.56	1.57e-5	4.67	8.24e-5	4.11
	1/80	5.36e-7	3.79	6.60e-7	4.58	1.90e-6	5.44
ARK(6,6,4)	1/160	5.61e-8	3.26	7.06e-8	3.22	1.86e-7	3.36
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	1.17e-4		1.57e-4		4.08e-4	
	1/40	1.22e-6	6.57	2.31e-6	6.08	8.56e-6	5.57
ARK(6,6,4)	1/80	3.18e-9	8.59	1.27e-8	7.50	9.49e-8	6.49
	1/160	2.00e-11	7.31	2.15e-11	9.21	5.45e-11	10.77

To test the ability of our method for more complex source terms, we replace the source term $u^2 + u$ in (6.1) by $-0.1 \exp(u) + \sin(u)$ while the initial and boundary conditions remain unchanged. In this case, the computational results with $\Delta x = 1/320$ are used as reference solutions. The results in Table 6.2 show that convergence orders are generally larger than those of the RK schemes. It should be noted that, for this complex source term, there are no analytical solutions to the algebra equation (3.5). To solve (3.5) numerically, we use the MATLAB function *fsolve* with the appropriate initial guess (see the details in Remark 3.3).

Example 2. Next we solve a 1D linear hyperbolic relaxation system

$$\begin{aligned} \partial_t u + \partial_x v &= 0, \\ \partial_t v + \partial_x u &= -\frac{1}{\epsilon}(u + v), \end{aligned} \quad (6.3)$$

in the domain $0 \leq x \leq 1$. The Jacobian matrix of (6.3) has two eigenvalues ± 1 and thus only one boundary condition should be imposed at $x = 0$ and also $x = 1$, respectively. We construct two sets of exact solutions to (6.3):

$$u(t, x) = \exp(t + x), \quad v(t, x) = -\exp(t + x) \quad (6.4)$$

and

$$u(t, x) = \exp(t + x) + \exp(-\frac{x}{\epsilon}), \quad v(t, x) = -\exp(t + x). \quad (6.5)$$

There exists no boundary layer for the first solution (6.4) while the boundary layer occurs at $x = 0$ for the second one (6.5) with small $\epsilon > 0$. The mesh is given by (3.2) and the solutions of u at $t = 1$ are used to evaluate the errors. In the second case, three interior grid points near the left boundary are excluded in the computation of the errors due to the existence of the boundary layer.

For the solution (6.4), we specify u at the left boundary and v at the right one, respectively, i.e.,

$$\begin{aligned} u(t, 0) &= \exp(t) \quad \text{at } x = 0, \\ v(t, 1) &= -\exp(t + 1) \quad \text{at } x = 1. \end{aligned} \quad (6.6)$$

Similarly, the boundary condition for (6.5) is given by

$$\begin{aligned} u(t, 0) &= \exp(t) + 1 \quad \text{at } x = 0, \\ v(t, 1) &= -\exp(t + 1) \quad \text{at } x = 1. \end{aligned} \quad (6.7)$$

The errors for the first case with solution (6.4) are given in Table 6.3, from which the desired second- and third-order convergence can be observed for the GSA(4,4,2) and GSA(4,4,3) schemes, respectively. Again, the convergence order for the ARK(6,6,4) scheme is around five since the error of the spatial discretization is much more dominant than that of the temporal one. For the second case with solution (6.5), we take $\epsilon = 1$ and 10^{-10} to test the convergence order. Table 6.4 shows that for the GSA(4,4,2) and GSA(4,4,3) schemes the present boundary treatment method produces designed accuracy for both choices of $\Delta x \ll \epsilon$ and $\Delta x \gg \epsilon$. For the ARK(6,6,3) scheme, the fifth-order convergence is achieved when $\epsilon = 1$ while the numerical solution does not converge to the exact solution for $\epsilon = 10^{-10} \ll \Delta x$. The latter may be due to the fact that the ARK(6,6,3) scheme does not satisfy the GSA property. Fig. 6.1 shows the solutions near the left boundary for the

Table 6.3

Example 2: Error table for the linear system (6.3) with the solution (6.4).

	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	1.42e-4		1.60e-4		3.62e-4	
	1/40	3.72e-5	1.93	4.08e-5	1.97	5.67e-5	2.68
	1/80	9.66e-6	1.95	1.06e-5	1.95	1.46e-5	1.96
	1/160	2.44e-6	1.98	2.68e-6	1.99	3.65e-6	2.00
	1/320	6.12e-7	2.00	1.67e-7	2.00	9.10e-7	2.00
GSA(4,4,3)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	9.77e-4		1.06e-3		2.37e-3	
	1/40	1.08e-4	3.03	1.25e-4	3.08	3.20e-4	2.89
	1/80	1.46e-5	2.88	1.71e-5	2.87	6.59e-5	2.28
	1/160	1.78e-6	3.03	1.95e-6	3.14	5.58e-6	3.56
ARK(6,6,4)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	2.66e-7		2.92e-7		5.45e-7	
	1/40	8.58e-9	4.96	9.52e-9	4.94	1.90e-8	4.84
	1/80	2.78e-10	4.95	3.09e-10	4.95	6.12e-10	4.96
	1/160	9.10e-12	4.93	1.01e-11	4.94	1.96e-11	4.96

Table 6.4

Example 2: Error table for the linear system (6.3) with the solution (6.5).

$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	1.27e-4		1.36e-4		1.74e-4	
	1/40	2.82e-5	2.17	2.96e-5	2.20	3.74e-5	2.22
	1/80	5.74e-6	2.30	6.05e-6	2.29	7.81e-6	2.26
	1/160	1.21e-6	2.24	1.29e-6	2.23	1.70e-6	2.20
	1/320	2.70e-7	2.16	2.91e-7	2.15	3.89e-7	2.13
$\epsilon = 10^{-10}$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	2.23e-4		4.26e-4		1.76e-3	
	1/40	4.93e-5	2.18	5.93e-5	2.85	9.31e-5	4.24
	1/80	1.46e-5	1.76	1.67e-5	1.83	3.16e-5	1.56
	1/160	3.87e-6	1.91	4.29e-6	1.96	8.80e-6	1.84
	1/320	9.91e-7	1.96	1.08e-6	1.99	2.18e-6	2.01
$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,3)	1/20	1.01e-3		1.18e-3		2.39e-3	
	1/40	1.20e-4	3.08	1.34e-4	3.14	3.17e-4	2.91
	1/80	1.57e-5	2.93	1.78e-5	2.91	6.49e-5	2.29
	1/160	1.90e-6	3.05	2.02e-6	3.14	5.52e-6	3.56
	1/320	2.49e-7	2.93	2.62e-7	2.95	7.19e-7	2.94
$\epsilon = 10^{-10}$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,3)	1/20	9.66e-4		1.16e-3		2.38e-3	
	1/40	1.16e-4	3.06	1.32e-4	3.13	3.22e-4	2.89
	1/80	1.53e-5	2.93	1.77e-5	2.90	6.58e-5	2.29
	1/160	1.84e-6	3.04	2.01e-6	3.14	5.72e-6	3.52
	1/320	2.51e-7	2.88	2.68e-7	2.91	7.07e-7	3.02
$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
ARK(6,6,4)	1/20	2.66e-7		2.92e-7		5.45e-7	
	1/40	8.58e-9	4.96	9.52e-9	4.94	1.90e-8	4.84
	1/80	2.78e-10	4.95	3.09e-10	4.95	6.12e-10	4.96
	1/160	9.10e-12	4.93	1.01e-11	4.94	1.96e-11	4.96

GSA(4,4,3) scheme with different ϵ and fixed $\Delta x = 1/80$ (the results for the GSA(4,4,2) scheme are similar). It can be seen that our method, equipped with the GSA schemes, behaves well even in the presence of boundary layer.

Example 3. We further construct the following nonlinear system

$$\begin{aligned} \partial_t u + \partial_x v &= 0, \\ \partial_t v + \partial_x(u + \frac{1}{2}(u + v)^2) &= -\frac{1}{\epsilon}(u + v + (u + v)^2), \end{aligned} \quad (6.8)$$

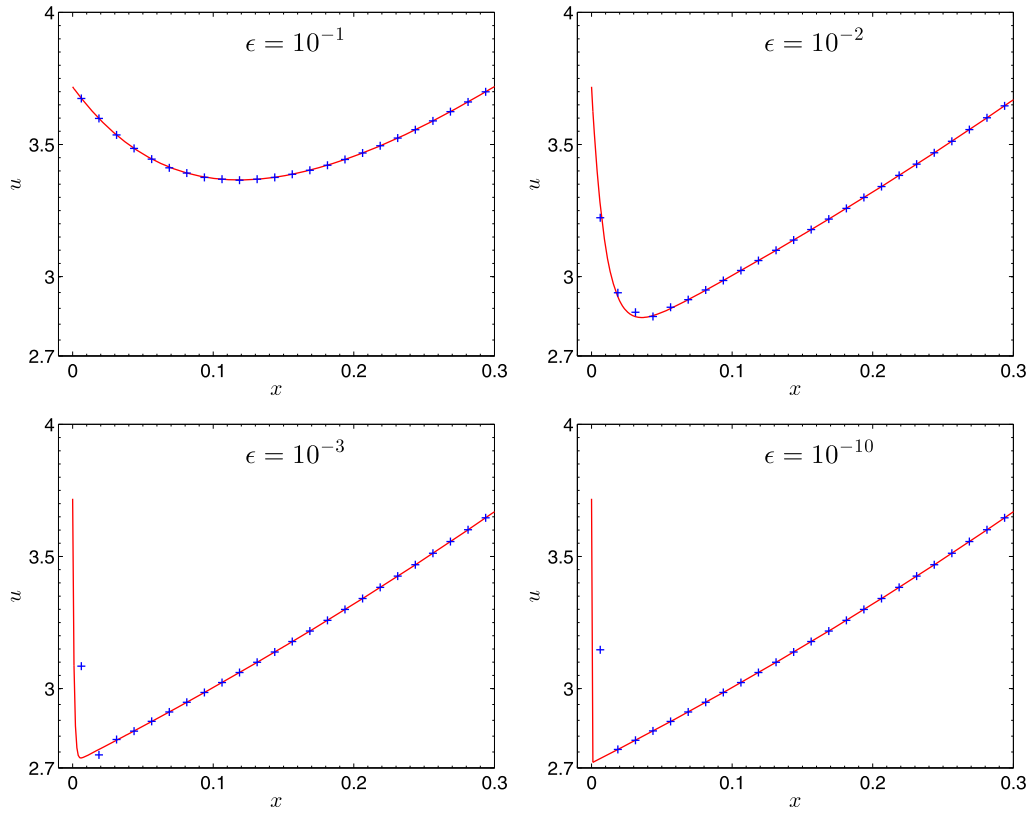


Fig. 6.1. Example 2: Comparison of the numerical solution (markers) and analytical solution (line) of the linear system (6.3) with the solution (6.5) for the GSA(4,4,3) scheme with different ϵ and fixed $\Delta x = 1/80$.

in the domain $0 \leq x \leq 1$ to test the accuracy of our method. The expressions (6.4) and (6.5) are also the exact solutions to (6.8) if cooperated with appropriate initial and boundary conditions. The Jacobian matrix in (6.8) is

$$A = \begin{pmatrix} 0 & 1 \\ 1+u+v & u+v \end{pmatrix},$$

which can be diagonalized as

$$LAR = \begin{pmatrix} 1+u+v & 0 \\ 0 & -1 \end{pmatrix},$$

where $1+u+v > 0$ and

$$L = \begin{pmatrix} 1 & 1 \\ 1 & -\frac{1}{1+u+v} \end{pmatrix}, \quad R = L^{-1} = \begin{pmatrix} \frac{1}{2+u+v} & \frac{1+u+v}{2+u+v} \\ \frac{1+u+v}{2+u+v} & -\frac{1+u+v}{2+u+v} \end{pmatrix}.$$

The boundary conditions in (6.6) and (6.7) are employed here.

The errors of the solution (6.4) with boundary condition (6.6) are given in Table 6.5 and desired convergence orders are achieved again. Similar to the linear system, we take $\epsilon = 1$ and 10^{-10} for the solution (6.5) with boundary condition (6.7) and the errors are shown in Table 6.6. It is clear that for both choices of ϵ the convergence orders are about two and three, for GSA(4,4,2) and GSA(4,4,3), respectively. Similar to Example 2, the ARK(6,6,3) scheme loses its stability for $\epsilon = 10^{-10}$ while fifth-order convergence is observed when $\epsilon = 1$. These results demonstrate the effectiveness of our method for nonlinear systems.

6.2. 2D reactive Euler equations

All the following examples are devoted to validate our boundary treatment for the 2D reactive Euler equations (5.1). The parameters are fixed as in [31]:

Table 6.5

Example 3: Error table for the nonlinear system (6.8) with solution (6.4).

	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	1.42e-4		1.60e-4		3.62e-4	
	1/40	3.72e-5	1.93	4.08e-5	1.97	5.67e-5	2.68
	1/80	9.66e-6	1.95	1.06e-5	1.94	1.46e-5	1.96
	1/160	2.44e-6	1.98	2.68e-6	1.99	3.65e-6	2.00
	1/320	6.12e-7	2.00	6.69e-7	2.00	9.10e-7	2.00
GSA(4,4,3)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	9.25e-4		1.09e-3		2.39e-3	
	1/40	1.13e-4	3.03	1.29e-4	3.08	3.22e-4	2.89
	1/80	1.52e-5	2.89	1.76e-5	2.87	6.61e-5	2.28
	1/160	1.86e-6	3.03	2.01e-6	3.13	5.61e-6	3.56
ARK(6,6,4)	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
	1/20	2.05e-7		2.47e-7		5.37e-7	
	1/40	6.93e-9	4.88	8.37e-9	4.88	1.87e-8	4.85
	1/80	2.31e-10	4.91	2.76e-10	4.92	6.01e-10	4.96
	1/160	7.69e-12	4.91	9.08e-12	4.93	1.94e-11	4.96

Table 6.6

Example 3: Error table for the nonlinear system (6.8) with solution (6.5).

$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	2.64e-4		3.13e-4		4.41e-4	
	1/40	4.74e-5	2.56	5.25e-5	2.58	7.18e-5	2.62
	1/80	7.46e-6	2.67	7.99e-6	2.72	1.05e-5	2.78
	1/160	7.86e-7	2.69	1.22e-6	2.71	1.53e-6	2.77
	1/320	9.25e-8	2.61	2.03e-7	2.60	2.68e-7	2.51
$\epsilon = 10^{-10}$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,2)	1/20	2.23e-4		4.26e-4		1.76e-3	
	1/40	4.93e-5	2.18	5.93e-5	2.85	9.31e-5	4.24
	1/80	1.46e-5	1.76	1.67e-5	1.83	3.16e-5	1.56
	1/160	3.87e-6	1.91	4.29e-6	1.96	8.80e-6	1.84
	1/320	9.91e-7	1.96	1.08e-6	1.99	2.18e-6	2.01
$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,3)	1/20	2.52e-4		2.99e-4		4.33e-4	
	1/40	4.28e-5	2.56	4.77e-5	2.65	6.74e-5	2.68
	1/80	6.09e-6	2.81	6.58e-6	2.86	8.98e-6	2.91
	1/160	7.86e-7	2.95	8.33e-7	2.98	1.10e-6	3.03
	1/320	9.25e-8	3.09	9.66e-8	3.11	1.22e-7	3.18
$\epsilon = 10^{-10}$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
GSA(4,4,3)	1/20	7.97e-4		1.14e-3		2.90e-3	
	1/40	1.07e-4	2.90	1.35e-4	3.08	3.86e-4	2.91
	1/80	1.40e-5	2.93	1.69e-5	3.00	6.61e-5	2.55
	1/160	1.76e-6	2.99	1.95e-6	3.11	5.60e-6	3.56
	1/320	2.34e-7	2.91	2.55e-7	2.94	7.27e-7	2.95
$\epsilon = 1$							
	Δx	L^1 error	order	L^2 error	order	L^∞ error	order
ARK(6,6,4)	1/20	1.38e-7		1.51e-7		2.30e-7	
	1/40	8.38e-9	4.04	8.76e-9	4.11	1.23e-8	4.22
	1/80	2.01e-9	5.38	2.17e-10	5.33	3.15e-10	5.23
	1/160	5.30e-12	5.25	5.66e-12	5.26	9.03e-12	5.12

$$\gamma = 1.2, \quad q = 50, \quad \tilde{T} = 50, \quad \tilde{K} = 2566.4.$$

For simplicity, we only use the third-order GSA(4,4,3) scheme for 2D problems.

Example 4. We first verify the accuracy of the method through a problem with the exact solution

$$\rho = 1 + 0.3 \sin(2\pi(x + y - t)), \quad u = 1, \quad v = 0, \quad p = 1, \quad Y = 0 \quad (6.9)$$

in a square domain $[0, 1] \times [0, 1]$, subject to periodic boundary conditions at the left and right boundaries and $v = 0$ at the top and bottom boundaries. The latter boundary condition is treated with our method. The mesh for computation is

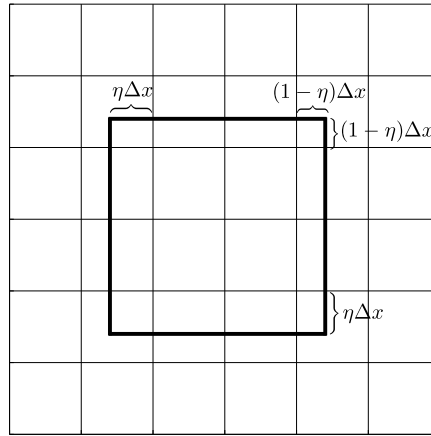


Fig. 6.2. Example 4: Illustration of the mesh. The thin lines are the grid lines and the thick lines are the boundaries of the square domain $[0, 1] \times [0, 1]$. The distance between the bottom boundary and its adjacent grid line is $\eta\Delta x$ and that at the top boundary is $(1 - \eta)\Delta x$. In a similar way the left and right boundaries are placed. Two layers of grid points are required outside each boundary for the third-order WENO scheme.

Table 6.7

Example 4: Error table for 2D reactive Euler equations with solution (6.9). The velocity v is prescribed at the top and bottom boundaries.

$\eta = 0.5$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.37e-2		1.64e-2		3.18e-2	
1/40	4.12e-3	1.73	5.55e-3	1.56	1.29e-2	1.30
1/80	7.78e-4	2.40	1.27e-3	2.13	3.75e-3	1.78
1/160	7.35e-5	3.40	1.35e-4	3.23	5.25e-4	2.84
$\eta = 0.7$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.41e-2		1.64e-2		3.10e-2	
1/40	4.10e-3	1.78	5.56e-3	1.56	1.27e-2	1.28
1/80	7.79e-4	2.40	1.27e-3	2.13	3.76e-3	1.76
1/160	7.34e-5	3.41	1.35e-4	3.23	5.32e-4	2.82
$\eta = 0.9$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.44e-2		1.66e-2		3.08e-2	
1/40	4.12e-3	1.81	5.58e-3	1.57	1.26e-2	1.28
1/80	7.70e-4	2.42	1.27e-3	2.14	3.69e-3	1.78
1/160	7.34e-5	3.39	1.35e-4	3.23	5.27e-4	2.81

illustrated in Fig. 6.2. As shown in the figure, the distance between the bottom boundary and its adjacent grid line is $\eta\Delta x$ and that at the top boundary is $(1 - \eta)\Delta x$. The left and right boundaries are placed in a similar way. We test the accuracy with $\eta = 0.5, 0.7$ and 0.9 , and the results are listed in Table 6.7. It can be seen that the influence of η on the errors is slight and the convergence order is about three for different choices of η . These demonstrate the third-order accuracy of our method for 2D problems.

Next, we prescribe the boundary condition on the density ρ , instead of v , at the top and bottom boundaries so that the traditional reflective boundary treatment fails. With this boundary condition, the errors are listed in Table 6.8, from which third-order convergence can be observed again for different η . This demonstrates the advantage of our boundary treatment for general types of boundary conditions.

Example 5. Next we test our method with the example in [31] containing discontinuity. The computational domain is $[0, 2] \times [0, 2]$ and the initial data is

$$(\rho, u, v, p, Y) = \begin{cases} (1, 0, 0, 80, 0.2), & x^2 + y^2 \leq 0.36, \\ (1, 0, 0, 10, 0.8), & \text{otherwise,} \end{cases} \quad (6.10)$$

where p and Y are different from that in [31] to avoid possibly negative pressure or density. The boundary conditions for the left and bottom boundaries are $u = 0$ and $v = 0$, respectively, and the solutions of the grid points outside the right and top boundaries are equal to those at the grid points inside the domain.

Table 6.8

Example 4: Error table for 2D reactive Euler equations with solution (6.9). The density ρ is prescribed at the top and bottom boundaries.

$\eta = 0.5$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.33e-2		1.61e-2		3.44e-2	
1/40	4.04e-3	1.72	5.47e-3	1.55	1.41e-2	1.29
1/80	7.75e-4	2.38	1.26e-3	2.12	4.07e-3	1.79
1/160	7.62e-5	3.35	1.35e-4	3.22	5.44e-4	2.90
$\eta = 0.7$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.37e-2		1.62e-2		3.39e-2	
1/40	4.04e-3	1.76	5.49e-3	1.56	1.41e-2	1.26
1/80	7.75e-4	2.38	1.26e-3	2.13	4.13e-3	1.77
1/160	7.59e-5	3.35	1.35e-4	3.23	5.51e-4	2.90
$\eta = 0.9$						
Δx	L^1 error	order	L^2 error	order	L^∞ error	order
1/20	1.42e-2		1.65e-2		3.79e-2	
1/40	4.05e-3	1.81	5.53e-3	1.58	1.48e-2	1.36
1/80	7.66e-4	2.40	1.26e-3	2.13	4.20e-3	1.81
1/160	7.59e-5	3.34	1.35e-4	3.23	5.48e-4	2.94

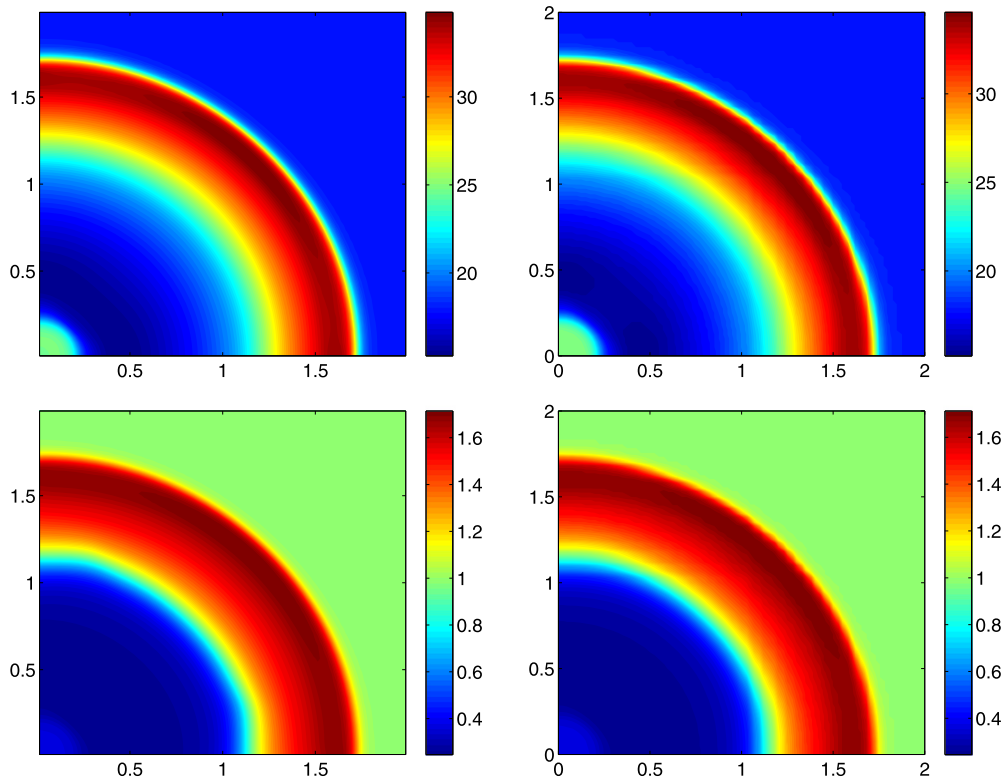


Fig. 6.3. Example 5: Contour plots of pressure (top) and density (bottom) of the present method (left) and the reflective boundary treatment in [21] (right) with $\Delta x = 1/40$ at $t = 0.16$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The mesh for computation is the same as that illustrated in Fig. 6.2 with a ratio η . We compare our results with those of the reflective boundary treatment (left and bottom) in [21] under the same mesh size $\Delta x = 1/40$. The contour plots of p and ρ at $t = 0.16$ are given in Fig. 6.3 and the solutions along the horizontal diagonal line $y = x$ are plotted in Fig. 6.4. Good agreement between the two methods can be observed. Additionally, we compare in Fig. 6.5 the solutions of different ratio η . The numerical results agree quite well with different values of η .

Example 6. We test the detonation diffraction in this example with the same configuration of boundaries in [31] (see Fig. 6.6). The spatial domain is $[0, 5] \times [0, 5]$ and the initial condition is

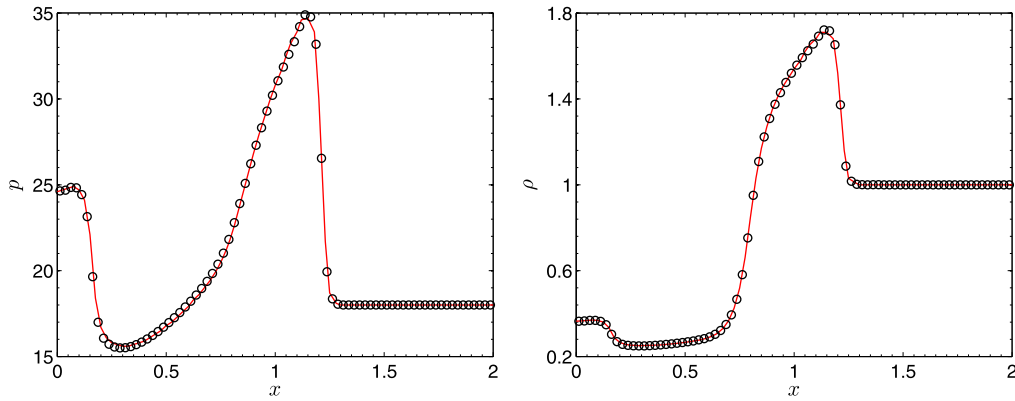


Fig. 6.4. Example 5: Comparison of the pressure (left) and density (right) of the present method (circle) and the reflective boundary treatment in [21] (line) along the diagonal line $y = x$.

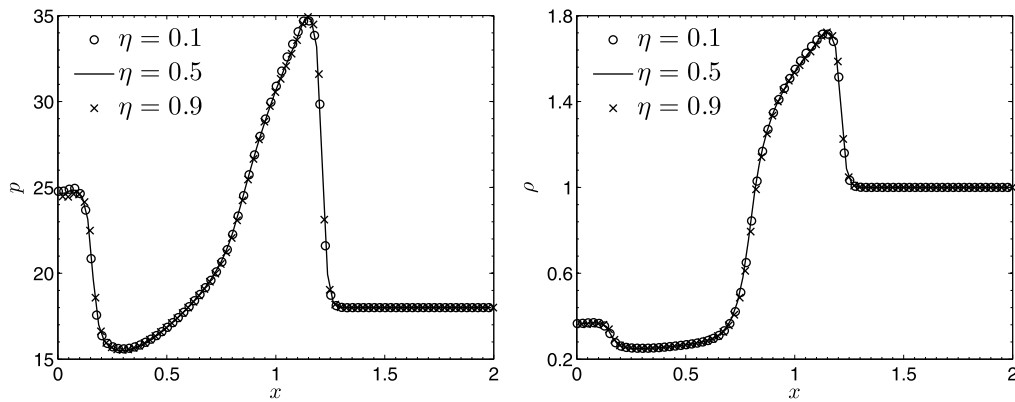


Fig. 6.5. Example 5: Comparison of the pressure (left) and density (right) of the present method with different η along the diagonal line $y = x$.

$$(\rho, u, v, E, Y) = \begin{cases} (11, 6.18, 0, 970, 1), & x < 0.5, \\ (5, 0, 0, 400, 1), & \text{otherwise,} \end{cases}$$

which is constructed based on that in [31] to avoid negative pressure or density. The boundary condition is $\mathbf{n} \cdot \mathbf{u} = 0$ with \mathbf{n} being the normal direction of the boundary, except that at $x = 0$, $(\rho, u, v, E, Y) = (11, 6.18, 0, 970, 1)$. The terminal time is $t = 0.6$ and the mesh size is $\Delta x = 1/20$. The ratio η is fixed as 0.5 for all the boundaries.

The results of the present boundary treatment and the reflective boundary treatment in [21] are presented in Fig. 6.6 (contour plot) and Fig. 6.7 (solutions along $y = 2.5$). It is clear that good agreements between the two different boundary treatment are achieved again.

Example 7. Finally, we test our method through an example with multiple obstacles in [21,20], which is designed following [31]. In this example, the spatial domain is $[0, 10] \times [0, 10]$ and there are two obstacles with positions $[1, 3] \times [0, 3]$ and $[5, 10] \times [0, 5]$, respectively. The initial condition is set as (6.10) and the boundary condition is $\mathbf{n} \cdot \mathbf{u} = 0$ for all the boundaries. We take $\Delta x = 1/20$. The results of the present method and the reflective boundary conditions in [21] at $t = 1$ and $t = 2$ are shown in Fig. 6.8 and Fig. 6.9, respectively. It can be observed that the results of the two methods are very similar even with the very coarse mesh, which demonstrate the effectiveness of our boundary treatment for complex domains.

7. Conclusions and remarks

In this paper, we propose a high order finite difference boundary treatment method for the IMEX RK schemes solving hyperbolic systems with possibly stiff source terms on a Cartesian mesh. By combining the idea of using the RK schemes at the boundary and the ILW procedure, our method not only preserves the accuracy of the RK schemes but also possesses good stability. Our method is different from the widely used approach for the explicit RK schemes by imposing boundary conditions for intermediate solutions [14], which could not be derived for the IMEX schemes. In addition, the intermediate

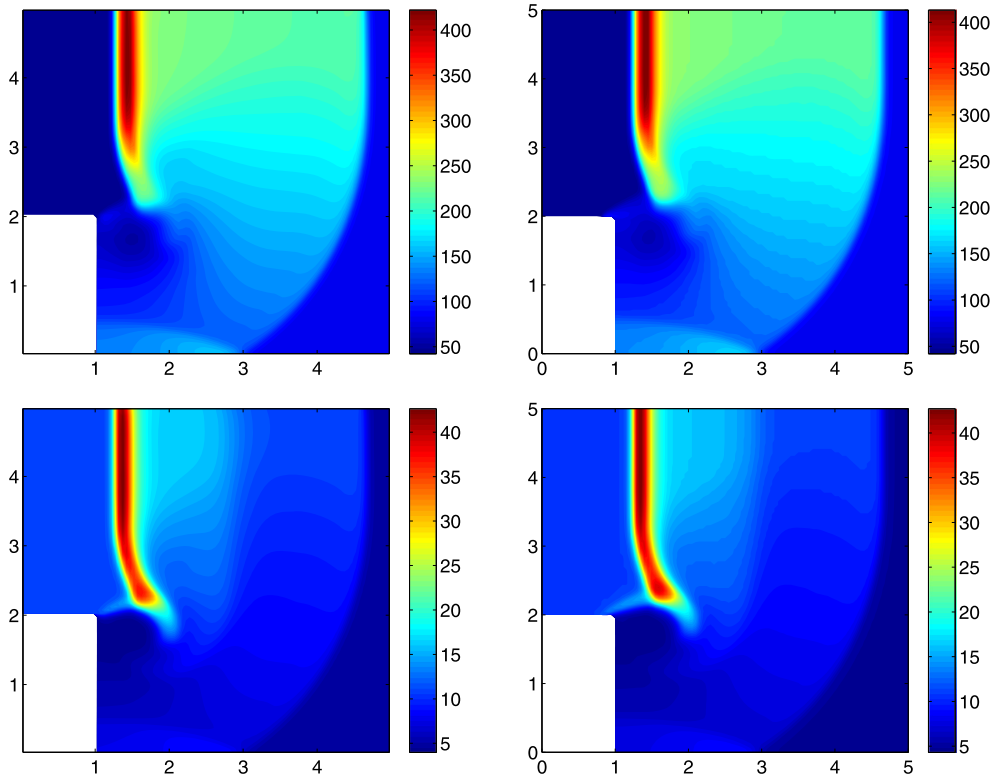


Fig. 6.6. Example 6: Contour plots of pressure (top) and density (bottom) of the present method (left) and the reflective boundary treatment in [21] (right) with $\Delta x = 1/20$ at $t = 0.6$.

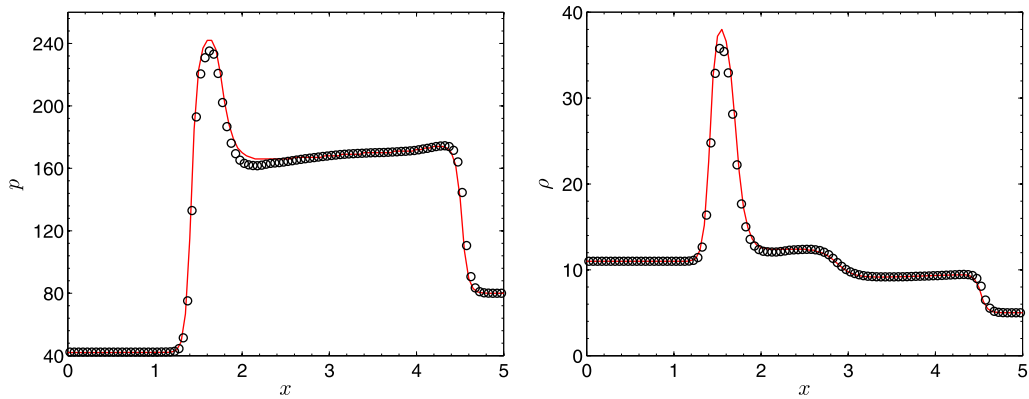


Fig. 6.7. Example 6: Comparison of the pressure (left) and density (right) of the present method (circle) and the reflective boundary treatment in [21] (line) along the horizontal center line $y = 2.5$.

boundary conditions are only available for explicit RK schemes up to third order while our method applies to arbitrary order IMEX and explicit RK schemes. Furthermore, the accuracy of our method is justified for a specific third-order IMEX scheme solving the scalar hyperbolic equation with source term. Finally, both 1D examples and 2D reactive Euler equations are used to demonstrate the good stability and desired accuracy of our boundary treatment for three specific second-, third- and fourth-order IMEX schemes.

Since the IMEX RK schemes include explicit schemes as special cases, the present boundary treatment also applies to explicit RK schemes. It does not rely on intermediate-stage boundary conditions and thus is expected to be valid for explicit RK schemes of order higher than three. This is investigated in a separate work [36]. Moreover, our method may be adapted to IMEX RK schemes solving the other partial differential equations, e.g. convection-diffusion equations [15,25,32,33] and the Boltzmann equation [16,19,18,23]. Many unresolved issues for the boundary treatment of RK schemes might be explored based on the idea of this work.

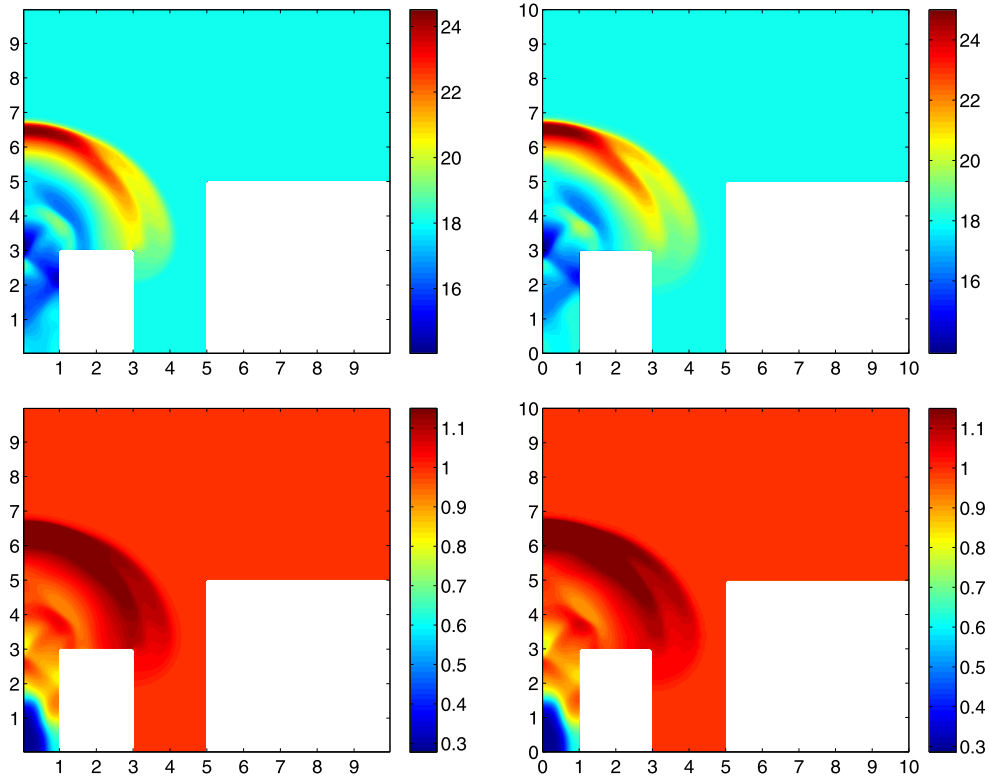


Fig. 6.8. Example 7: Contour plots of pressure (top) and density (bottom) of the present method (left) and the reflective boundary treatment in [21] (right) with $\Delta x = 1/20$ at $t = 1$.

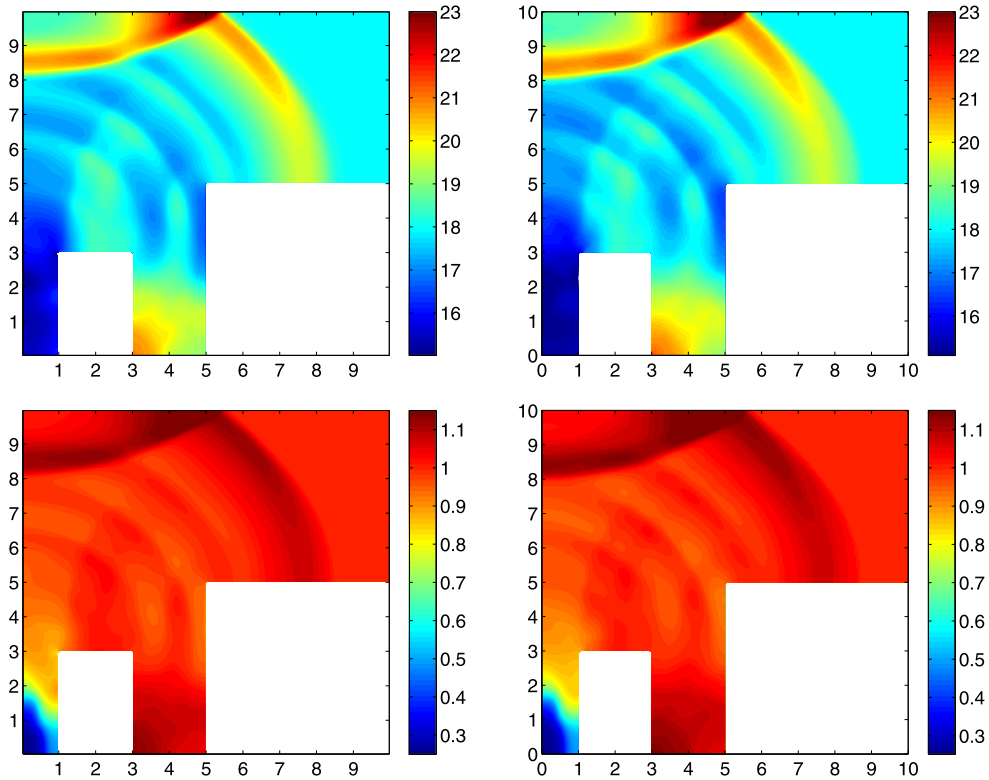


Fig. 6.9. Example 7: Contour plots of pressure (top) and density (bottom) of the present method (left) and the reflective boundary treatment scheme in [21] (right) with $\Delta x = 1/20$ at $t = 2$.

CRediT authorship contribution statement

Weifeng Zhao: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft, Writing - review & editing. **Juntao Huang:** Conceptualization, Formal analysis, Investigation, Methodology, Validation, Writing - original draft, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Professor Wen-An Yong for suggesting us to consider the current problem and for his important advice. We would also like to express our thanks to the referees for providing insightful comments which substantially helped improving the quality of the paper. This work was supported by the National Natural Science Foundation of China (No. 11801030, No. 11861131004).

References

- [1] S. Abarbanel, D. Gottlieb, M.H. Carpenter, On the removal of boundary errors caused by Runge–Kutta integration of nonlinear partial differential equations, *SIAM J. Sci. Comput.* 17 (3) (1996) 777–782.
- [2] I. Alonso-Mallo, Runge–Kutta methods without order reduction for linear initial boundary value problems, *Numer. Math.* 91 (4) (2002) 577–603.
- [3] I. Alonso-Mallo, B. Cano, Spectral/Rosenbrock discretizations without order reduction for linear parabolic problems, *Appl. Numer. Math.* 41 (2) (2002) 247–268.
- [4] I. Alonso-Mallo, B. Cano, M. Moreta, Order reduction and how to avoid it when explicit Runge–Kutta–Nyström methods are used to solve linear partial differential equations, *J. Comput. Appl. Math.* 176 (2) (2005) 293–318.
- [5] I. Alonso-Mallo, B. Cano, N. Reguera, Avoiding order reduction when integrating linear initial boundary value problems with Lawson methods, *IMA J. Numer. Anal.* 37 (4) (2016) 2091–2119.
- [6] I. Alonso-Mallo, B. Cano, N. Reguera, Avoiding order reduction when integrating linear initial boundary value problems with exponential splitting methods, *IMA J. Numer. Anal.* 38 (3) (2017) 1294–1323.
- [7] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.* 25 (2–3) (1997) 151–167.
- [8] S. Boscarino, On an accurate third order implicit-explicit Runge–Kutta method for stiff problems, *Appl. Numer. Math.* 59 (2005) 1515–1528.
- [9] S. Boscarino, L. Pareschi, On the asymptotic properties of IMEX Runge–Kutta schemes for hyperbolic balance laws, *J. Comput. Appl. Math.* 316 (2017) 60–73.
- [10] S. Boscarino, L. Pareschi, G. Russo, Implicit-explicit Runge–Kutta schemes for hyperbolic systems and kinetic equations in the diffusion limit, *SIAM J. Sci. Comput.* 35 (1) (2013) A22–A51.
- [11] S. Boscarino, L. Pareschi, G. Russo, A unified IMEX Runge–Kutta approach for hyperbolic systems with multiscale relaxation, *SIAM J. Numer. Anal.* 55 (4) (2017) 2085–2109.
- [12] S. Boscarino, J.-M. Qiu, G. Russo, Implicit-explicit integral deferred correction methods for stiff problems, *SIAM J. Sci. Comput.* 40 (2) (2018) A787–A816.
- [13] S. Boscarino, G. Russo, On a class of uniformly accurate IMEX Runge–Kutta schemes and applications to hyperbolic systems with relaxation, *SIAM J. Sci. Comput.* 31 (3) (2009) 1926–1945.
- [14] M.H. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge–Kutta time discretizations for the initial boundary value problem: a study of the boundary error, *SIAM J. Sci. Comput.* 16 (6) (1995) 1241–1252.
- [15] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, *Appl. Numer. Math.* 44 (1) (2003) 139–181.
- [16] F. Filbet, C. Yang, An inverse Lax–Wendroff method for boundary conditions applied to Boltzmann type models, *J. Comput. Phys.* 245 (19) (2013) 43–61.
- [17] B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems, *Math. Comput.* 29 (1975) 396–406.
- [18] J. Hu, R. Shu, X. Zhang, Asymptotic-preserving and positivity-preserving implicit-explicit schemes for the stiff BGK equation, *SIAM J. Numer. Anal.* 56 (2) (2018) 942–973.
- [19] J. Hu, X. Zhang, On a class of implicit-explicit Runge–Kutta schemes for stiff kinetic equations preserving the Navier–Stokes limit, *J. Sci. Comput.* 73 (2–3) (2017) 797–818.
- [20] J. Huang, C.-W. Shu, Positivity-preserving time discretizations for production–destruction equations with applications to non-equilibrium flows, *J. Sci. Comput.* 78 (3) (2019) 1811–1839.
- [21] J. Huang, W. Zhao, C.-W. Shu, A third-order unconditionally positivity-preserving scheme for production–destruction equations with applications to non-equilibrium flows, *J. Sci. Comput.* 79 (2) (2019) 1015–1056.
- [22] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1) (1996) 202–228.
- [23] S. Jin, H. Lu, L. Pareschi, Efficient stochastic asymptotic-preserving implicit-explicit methods for transport equations with diffusive scalings and random inputs, *SIAM J. Sci. Comput.* 40 (2) (2018) A671–A696.
- [24] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1) (1994) 200–212.
- [25] J. Lu, J. Fang, S. Tan, C.-W. Shu, M. Zhang, Inverse Lax–Wendroff procedure for numerical boundary conditions of convection–diffusion equations, *J. Comput. Phys.* 317 (2016) 276–300.
- [26] L. Pareschi, G. Russo, Implicit-explicit Runge–Kutta schemes for stiff systems of differential equations, *Recent Trends Numer. Anal.* 3 (2000) 269–289.
- [27] L. Pareschi, G. Russo, Implicit-explicit Runge–Kutta schemes and applications to hyperbolic systems with relaxation, *J. Sci. Comput.* 25 (1) (2005) 129–155.
- [28] D. Pathria, The correct formulation of intermediate boundary conditions for Runge–Kutta time integration of initial boundary value problems, *SIAM J. Sci. Comput.* 18 (5) (1997) 1255–1266.
- [29] S. Tan, C.-W. Shu, Inverse Lax–Wendroff procedure for numerical boundary conditions of conservation laws, *J. Comput. Phys.* 229 (21) (2010) 8144–8166.
- [30] S. Tan, C. Wang, C.-W. Shu, J. Ning, Efficient implementation of high order inverse Lax–Wendroff boundary treatment for conservation laws, *J. Comput. Phys.* 231 (6) (2012) 2510–2527.

- [31] C. Wang, X. Zhang, C.-W. Shu, J. Ning, Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations, *J. Comput. Phys.* 231 (2) (2012) 653–665.
- [32] H. Wang, C.-W. Shu, Q. Zhang, Stability and error estimates of local discontinuous Galerkin methods with implicit-explicit time-marching for advection-diffusion problems, *SIAM J. Numer. Anal.* 53 (1) (2015) 206–227.
- [33] H. Wang, Q. Zhang, C.-W. Shu, Third order implicit-explicit Runge-Kutta local discontinuous Galerkin methods with suitable boundary treatment for convection-diffusion problems with Dirichlet boundary conditions, *J. Comput. Appl. Math.* 342 (2018) 164–179.
- [34] W.-A. Yong, Singular perturbations of first-order hyperbolic systems with stiff source terms, *J. Differ. Equ.* 155 (1) (1999) 89–132.
- [35] W.-A. Yong, Basic aspects of hyperbolic relaxation systems, in: H. Freisthler, A. Szepessy (Eds.), *Advances in the Theory of Shock Waves*, Progress in Nonlinear Differential Equations and Their Applications, Birkhäuser, Boston, MA, 2001, pp. 259–305.
- [36] W. Zhao, J. Huang, S.J. Ruuth, Boundary treatment of high order Runge-Kutta methods for hyperbolic conservation laws, *J. Comput. Phys.* 421 (2020) 109697.