Journal of Computational Physics 229 (2010) 8144–8166

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws ☆

Sirui Tan, Chi-Wang Shu *

Division of Applied Mathematics, Brown University, Providence, RI 02912, United States

ARTICLE INFO

Article history:
Received 6 April 2010
Received in revised form 6 July 2010
Accepted 9 July 2010
Available online 7 August 2010

Keywords:
Numerical boundary conditions
Hyperbolic conservation laws
Finite difference method
Cartesian mesh
Lax-Wendroff procedure
Extrapolation
Solid wall

ABSTRACT

We develop a high order finite difference numerical boundary condition for solving hyperbolic conservation laws on a Cartesian mesh. The challenge results from the wide stencil of the interior high order scheme and the fact that the boundary intersects the grids in an arbitrary fashion. Our method is based on an inverse Lax-Wendroff procedure for the inflow boundary conditions. We repeatedly use the partial differential equation to write the normal derivatives to the inflow boundary in terms of the time derivatives and the tangential derivatives. With these normal derivatives, we can then impose accurate values of ghost points near the boundary by a Taylor expansion. At outflow boundaries, we use Lagrange extrapolation or least squares extrapolation if the solution is smooth, or a weighted essentially non-oscillatory (WENO) type extrapolation if a shock is close to the boundary. Extensive numerical examples are provided to illustrate that our method is high order accurate and has good performance when applied to one and two-dimensional scalar or system cases with the physical boundary not aligned with the grids and with various boundary conditions including the solid wall boundary condition. Additional numerical cost due to our boundary treatment is discussed in some of the examples.

1. Introduction

In this paper, we develop a high order boundary treatment for solving strongly hyperbolic conservation laws with high order finite difference methods on a Cartesian mesh. In a computational domain with complex geometries, a Cartesian mesh makes the numerical method efficient and easy to implement compared with boundary fitted structured or unstructured meshes. However, there are two kinds of difficulties imposing inflow boundary conditions. First, a high order interior scheme needs a suitable treatment for *several* ghost points near the boundary because of the wide numerical stencil. Secondly, the grid points are usually not located on the physical boundary when using a Cartesian mesh. In other words, the boundary intersects the grids in an arbitrary fashion. In finite volume methods, this leads to a severe time step restriction. The so-called *h*-box method is developed to overcome this problem, see [1] and the references therein. In solving compressible inviscid Euler equations, the most popular way to impose the no-penetration boundary condition at solid walls is the reflection technique, where ghost points are added behind the wall. All interior solution components are reflected symmetrically to values of ghost points except for the normal velocity whose sign is reversed. This method works well if the grid points are symmetrically located with respect to the wall and leads to large errors otherwise. An accurate implementation of solid wall

☆ Research supported by AFOSR Grant FA9550-09-1-0126 and NSF Grant DMS-0809086.
* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: sirui@dam.brown.edu (S. Tan), shu@dam.brown.edu (C.-W. Shu).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.07.014

boundary conditions in curved geometries is developed for discontinuous Galerkin methods on unstructured, straight-sided element meshes in [12].

A Cartesian embedded boundary method based on the finite difference formulation is developed to solve the wave equation with Dirichlet or Neumann boundary conditions in [9–11] and hyperbolic conservation laws in [17]. The technique used in [17] is formally second order accurate with an explicit time step determined by the grid size away from the boundary. It is essentially based on a three point interior scheme so that only points just outside the boundary are ghost points, which are imposed by extrapolation. The feasibility and effectiveness to generalize this approach to higher order remain to be demonstrated.

The idea of the procedure developed in this paper comes from [7], in which a Lax-Wendroff type boundary condition procedure is introduced for solving static Hamilton–Jacobi equations with a third order method. Later, Xiong et al. extend it to fifth order for the same type of problems in [20]. This approach is based on repeatedly using the partial differential equation (PDE) to write the normal derivatives to the inflow boundary in terms of the tangential derivatives of the given boundary condition. With these normal derivatives, we can obtain accurate values of ghost points by a Taylor expansion from a point located on the boundary.

In this paper, we systematically extend this procedure to solve time dependent hyperbolic conservation laws. The difference with [7,20] mainly relies on three points. First, for time dependent problems, our boundary treatment procedure is in essence repeatedly using the PDE to convert normal spatial derivatives to tangential and time derivatives of the given boundary condition. Secondly, in systems of conservation laws, the inflow and outflow boundary conditions are coupled. As a result, special care must be taken for imposing outflow boundary conditions. We use high order Lagrange extrapolation or least squares extrapolation if the solution is smooth. In the presence of shocks near the outflow boundary, we develop a weighted essentially non-oscillatory (WENO) type extrapolation to prevent oscillations and maintain accuracy. Finally, this method should work well for solid wall boundary conditions. Our boundary treatment procedure, which uses repeatedly the PDE to convert normal spatial derivatives to tangential and time derivatives of the given boundary condition, is in some sense an inverse to the usual Lax-Wendroff procedure [13], in which the PDE is repeated used to convert time derivatives to spatial derivatives when discretizing the PDE in time with high order accuracy. We therefore refer to our method as the inverse Lax-Wendroff procedure.

This paper is organized as follows. In Section 2, we first give an overview of the discretization of the problem. Then we illustrate the idea of the inverse Lax-Wendroff procedure by one-dimensional scalar conservation laws. We will see a need for robust extrapolation if there is a shock near the outflow boundary. We develop a WENO type extrapolation for this purpose. The linear stability of our numerical boundary condition is shown afterwards. The method for one-dimensional scalar equation is then generalized to one-dimensional systems and further to two-dimensional problems. In Section 3, a variety of numerical examples are provided to demonstrate the effectiveness and generality of our approach. Concluding remarks are given in Section 4.

2. Scheme formulation

We consider strongly hyperbolic conservation laws for $\boldsymbol{U} = \boldsymbol{U}(x,y,t) \in \mathbb{R}^2$

$$\begin{cases} \boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y = 0 \quad (x,y) \in \Omega,\ t > 0, \\ \boldsymbol{U}(x,y,0) = \boldsymbol{U}_0(x,y) \quad (x,y) \in \bar{\Omega}, \end{cases} \tag{2.1}$$

on a bounded domain Ω with appropriate boundary conditions prescribed on $\partial\Omega$ at time t. We assume Ω is covered by a uniform Cartesian mesh $\Omega_h = \{(x_i,y_j)\colon\ 0 \leqslant i \leqslant N_x,\ 0 \leqslant j \leqslant N_y\}$ with mesh size $\Delta x = \Delta y$. The semi-discrete approximation of (2.1) is given by

$$\frac{d}{dt}\boldsymbol{U}_{ij}(t) = -\frac{1}{\Delta x}\left(\widehat{\boldsymbol{F}}_{i+1/2j} - \widehat{\boldsymbol{F}}_{i-1/2j}\right) - \frac{1}{\Delta y}\left(\widehat{\boldsymbol{G}}_{ij+1/2} - \widehat{\boldsymbol{G}}_{ij-1/2}\right), \tag{2.2}$$

where $\widehat{\boldsymbol{F}}_{i+1/2j}$ and $\widehat{\boldsymbol{G}}_{ij+1/2}$ are numerical fluxes.

We use a third order total variation diminishing (TVD) Runge–Kutta method [16] to integrate the system of ordinary differential equations (ODEs) (2.2) in time. For ease of notations, we suppose a system of initial value problems of ODEs is written as

$$u_t = L(u).$$

The third order TVD Runge–Kutta method is given by

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{2.3}$$

To make the accuracy in time compatible with the accuracy in space, we take $\Delta t = O(\Delta x^{\frac{s}{3}})$ if the space discretization is of order s, when performing the accuracy tests. Other time discretization techniques, such as the various strong stability preserving Runge–Kutta or multistep methods [4], could of course be used as well.

Special care must be taken when we impose *time dependent* boundary conditions in the two interior stages of the Runge–Kutta method (2.3). Suppose we have a time dependent boundary condition $g(t)$. The traditional match of time

$$\begin{aligned} &u^n \sim g(t_n),\\ &u^{(1)} \sim g(t_n + \Delta t),\\ &u^{(2)} \sim g(t_n + \Delta t/2), \end{aligned}$$

decreases the accuracy to second order as pointed out in [2]. It is shown in [2] that for hyperbolic systems (2.1) the following match of time maintains the third order accuracy of (2.3)

$$\begin{aligned} &u^n \sim g(t_n),\\ &u^{(1)} \sim g(t_n) + \Delta t g'(t_n),\\ &u^{(2)} \sim g(t_n) + \frac{1}{2}\Delta t g'(t_n) + \frac{1}{4}\Delta t^2 g''(t_n). \end{aligned} \tag{2.4}$$

For simplicity, we denote the boundary conditions for all stages at $t = t_n$ by $g(t_n)$, although $g(t_n)$ is actually different for each stage according to (2.4).

For definiteness, we use the fifth order finite difference WENO scheme with the Lax-Friedrichs flux splitting [8] to form the numerical fluxes $\widehat{\boldsymbol{F}}_{i+1/2,j}$ and $\widehat{\boldsymbol{G}}_{i,j+1/2}$ in (2.2) although our method is independent of the interior scheme. The fifth order WENO scheme requires a seven point stencil in both x and y directions, which is much wider compared with low order schemes. Near $\partial\Omega$ where the numerical stencil is partially outside of Ω, up to three ghost points are needed in each direction. We concentrate on how to define the values of $\boldsymbol{U}_{i,j}$ at ghost points in the rest of the paper.

2.1. One-dimensional scalar conservation laws

To best illustrate the idea of the inverse Lax-Wendroff type procedure, we use 1D scalar conservation laws as an example

$$\begin{cases} u_t + f(u)_x = 0 & x \in (-1,1), \quad t > 0,\\ u(-1,t) = g(t) & t > 0,\\ u(x,0) = u_0(x) & x \in [-1,1]. \end{cases} \tag{2.5}$$

We assume $f'(u(-1,t)) \geqslant \alpha > 0$ and $f'(u(1,t)) \geqslant \alpha > 0$ for $t > 0$. This assumption guarantees the left boundary $x = -1$ is an inflow boundary where a boundary condition is needed and the right boundary $x = 1$ is an outflow boundary where no boundary condition is needed.

Let us discretize the interval $(-1,1)$ by a uniform mesh

$$-1 + \Delta x/2 = x_0 < x_1 < \cdots < x_N = 1 - \Delta x/2. \tag{2.6}$$

Notice that both x_0 and x_N are not located on the boundary, which is chosen this way on purpose since it is usually not possible to align boundary with grid points in a two-dimensional domain with complex geometries. At the inflow boundary $x = -1$, a Taylor expansion of order $s-1$ gives

$$u(x_j,t_n) = \sum_{k=0}^{s-1} \frac{(x_j+1)^k}{k!} \left.\frac{\partial^k u}{\partial x^k}\right|_{x=-1,t=t_n} + O(\Delta x^s),$$

for $j = -1, -2, -3$. Hence a sth order approximation of the values u_j at the ghost points is

$$u_j = \sum_{k=0}^{s-1} \frac{(x_j+1)^k}{k!} \left.\frac{\partial^k u}{\partial x^k}\right|_{x=-1,t=t_n}, \quad j = -1,-2,-3. \tag{2.7}$$

Here we suppress the t_n dependence on the left hand side. We already have $u(-1,t_n) = g(t_n)$. To obtain the spatial derivatives, we utilize the PDE

$$u_t + f'(u)u_x = 0$$

and evaluate it at $x = -1$, $t = t_n$. We have

$$u_x(-1,t_n) = -\frac{u_t(-1,t_n)}{f'(u(-1,t_n))} = -\frac{g'(t_n)}{f'(g(t_n))},$$

where $f'(g(t_n))$ is bounded away from zero by the assumption that $x = -1$ is an inflow boundary. Differentiating the PDE with respect to time yields

$$u_{tt} + f''(u)u_t u_x + f'(u)u_{xt} = 0. \tag{2.8}$$

The term u_{xt} can be written as

$$u_{xt} = (u_t)_x = -(f'(u)u_x)_x = -f''(u)u_x^2 - f'(u)u_{xx}.$$

Substituting it into (2.8) we obtain an equation for u_{xx}

$$u_{tt} + f''(u)u_t u_x - f'(u)f''(u)u_x^2 - f'(u)^2 u_{xx} = 0. \tag{2.9}$$

Solving (2.9) for u_{xx} and evaluating it at $x = -1$, $t = t_n$, we have

$$u_{xx}(-1,t_n) = \frac{g''(t_n) + f''(g(t_n))g'(t_n)u_x(-1,t_n) - f'(g(t_n))f''(g(t_n))u_x^2(-1,t_n)}{f'(g(t_n))^2} = \frac{f'(g(t_n))g''(t_n) - 2f''(g(t_n))g'(t_n)^2}{f'(g(t_n))^3}.$$

Following the same procedure, we can obtain values of $\frac{\partial^k u}{\partial x^k}|_{x=-1,t=t_n}$, $k = 1,\ldots,s-1$.

The idea of converting time derivatives to spatial derivatives by repeatedly using the PDE comes from the original Lax-Wendroff scheme [13]. Since we convert spatial derivatives to time derivatives instead, our method is called the inverse Lax-Wendroff procedure. We remark that this procedure is independent of the interior scheme and the location of the boundary. The time derivatives can be obtained by either using the analytical derivatives of $g(t)$ if available or numerical differentiation. In the case of shocks going through the boundary, $g(t)$ is discontinuous. The stencil used for numerical differentiation should not contain any discontinuity, e.g. an essentially non-oscillatory (ENO) procedure [6] or a weighted ENO (WENO) procedure [8] can be used for this numerical differentiation.

At the outflow boundary $x = 1$, extrapolation of appropriate order is used. The sth order extrapolation is given by

$$\sum_{k=0}^{s} \binom{s}{k} (-1)^k u_{j-k} = 0, \quad j = N+1, N+2, N+3. \tag{2.10}$$

For example, the fifth order extrapolation is

$$u_j = u_{j-5} - 5u_{j-4} + 10u_{j-3} - 10u_{j-2} + 5u_{j-1}, \quad j = N+1, N+2, N+3.$$

An equivalent way to do the extrapolation is to use a Taylor expansion

$$u_j = \sum_{k=0}^{s-1} \frac{(x_j - 1)^k}{k!} u^{*(k)}, \tag{2.11}$$

where $u^{*(k)}$ is a $(s-k)$th order approximation of $\frac{\partial^k u}{\partial x^k}|_{x=1,t=t_n}$. Notice that (2.11) is consistent with (2.7) which is useful in the system case discussed later. If u is smooth near the boundary, $u^{*(k)}$ can be easily obtained by

$$u^{*(k)} = \left.\frac{d^k p_{s-1}(x)}{dx^k}\right|_{x=1}, \tag{2.12}$$

where $p_{s-1}(x)$ is a Lagrange polynomial of degree $s-1$ satisfying $p_{s-1}(x_{N-i}) = u_{N-i}$, $i = 0,\ldots,s-1$.

When a shock goes out of the boundary, there are not enough points between the shock and the boundary for high order extrapolation at a particular time. In this situation, high order extrapolation may lead to a severe oscillation near the shock. To prevent this from happening, we would like to have a lower order accurate but more robust extrapolation. The WENO type extrapolation is developed for this purpose.

2.2. One-dimensional WENO type extrapolation

Assume that we have a stencil of three points $x_0 = 0, x_1 = \Delta x, x_2 = 2\Delta x$ with point values u_j at x_j, $j = 0, 1, 2$. We aim to obtain a $(3-k)$th order approximation of $\frac{d^k u}{dx^k}|_{x=-\Delta x/2}$, which is denoted by $u^{*(k)}$, $k = 0, 1, 2$. We have three candidate substencils given by

$$S_r = \{x_0,\ldots,x_r\}, \quad r = 0,1,2.$$

On each substencil S_r, we can easily construct a Lagrange polynomial $p_r(x)$ of degree r

$$\begin{aligned} p_0(x) &= u_0, \\ p_1(x) &= \frac{u_1 - u_0}{\Delta x}x + u_0, \\ p_2(x) &= \frac{u_0 - 2u_1 + u_2}{2\Delta x^2}x^2 + \frac{-3u_0 + 4u_1 - u_2}{2\Delta x}x + u_0. \end{aligned}$$

Suppose $u(x)$ is smooth on S_2, $u^{*(k)}$ can be extrapolated by

$$u^{*(k)} = \sum_{r=0}^{2} d_r \frac{d^k p_r(x)}{dx^k}\bigg|_{x=-\Delta x/2},$$

where $d_0 = \Delta x^2$, $d_1 = \Delta x$, $d_2 = 1 - \Delta x - \Delta x^2$.

We now look for WENO type extrapolation in the form

$$u^{*(k)} = \sum_{r=0}^{2} \omega_r \frac{d^k p_r(x)}{dx^k}\bigg|_{x=-\Delta x/2}, \tag{2.13}$$

where ω_r are the nonlinear weights depending on the values of u_j. In the case that $u(x)$ is smooth in S_2, we would like to have

$$\begin{aligned} &\omega_0 = O(\Delta x^2),\\ &\omega_1 = O(\Delta x),\\ &\omega_2 = 1 - \omega_0 - \omega_1. \end{aligned} \tag{2.14}$$

This implies (2.13) is $(3-k)$th order accurate. Following the idea of the usual WENO procedure in [8], the nonlinear weights ω_r are chosen to be

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{2}\alpha_s}$$

with

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2},$$

where $\epsilon = 10^{-6}$ and β_r are the smoothness indicators, which are determined by

$$\beta_0 = \Delta x^2,$$

$$\beta_1 = \sum_{l=1}^{2}\int_{-\Delta x}^{0} \Delta x^{2l-1}\left(\frac{d^l}{dx^l}p_1(x)\right)^2 dx = (u_1 - u_0)^2, \tag{2.15}$$

$$\beta_2 = \sum_{l=1}^{2}\int_{-\Delta x}^{0} \Delta x^{2l-1}\left(\frac{d^l}{dx^l}p_2(x)\right)^2 dx = (61u_0^2 + 160u_1^2 + 74u_0u_2 + 25u_2^2 - 196u_1u_0 + 124u_1u_2)/12. \tag{2.16}$$

We next show that (2.14) is achieved by this choice of β_r. Taylor expansion of (2.15) and (2.16) at $x = 0$ gives

$$\beta_1 = u'^2\Delta x^2 + u'u''\Delta x^3 + \left(\frac{1}{4}u''^2 + \frac{1}{3}u'u'''\right)\Delta x^4 + O(\Delta x^5), \tag{2.17}$$

$$\beta_2 = u'^2\Delta x^2 - u'u''\Delta x^3 + \left(\frac{4}{3}u''^2 - \frac{5}{3}u'u'''\right)\Delta x^4 + O(\Delta x^5). \tag{2.18}$$

We omit ϵ in the expression of α_r and do the following estimates with the assumption that $C_1 = (u')^4 \neq 0$.

$$\alpha_0 = \frac{\Delta x^2}{\Delta x^4},$$

$$\alpha_1 = \frac{\Delta x}{\Delta x^4(C_1 + O(\Delta x))} = \frac{\Delta x}{\Delta x^4}\left(\frac{1}{C_1} + O(\Delta x)\right),$$

$$\alpha_2 = \frac{1 + O(\Delta x)}{\Delta x^4}\left(\frac{1}{C_1} + O(\Delta x)\right);$$

$$\omega_0 = \frac{\Delta x^2}{\Delta x^2 + \Delta x\left(\frac{1}{C_1} + O(\Delta x)\right) + (1 + O(\Delta x))\left(\frac{1}{C_1} + O(\Delta x)\right)} = \frac{\Delta x^2}{\frac{1}{C_1} + O(\Delta x)} = C_1\Delta x^2 + O(\Delta x^3),$$

$$\omega_1 = \frac{\Delta x\left(\frac{1}{C_1} + O(\Delta x)\right)}{\frac{1}{C_1} + O(\Delta x)} = \Delta x + O(\Delta x^2).$$

If S_1 contains a discontinuity, we have $\beta_1 = O(1)$ and $\beta_2 = O(1)$. $u^{*(0)}$ reduces to a first order approximation since

$$\alpha_0 = \frac{1}{\Delta x^2},$$
$$\alpha_1 = O(\Delta x),$$
$$\alpha_2 = O(1);$$

$$\omega_1 = \frac{O(\Delta x)}{\frac{1}{\Delta x^2} + O(\Delta x) + O(1)} = O(\Delta x^3),$$

$$\omega_2 = \frac{O(1)}{\frac{1}{\Delta x^2} + O(\Delta x) + O(1)} = O(\Delta x^2).$$

Notice that as $\Delta x \to 0$ the weights assigned to the non-smooth stencils S_1 and S_2 vanish in a rate of $O(\Delta x^3)$ and $O(\Delta x^2)$, respectively. If only S_2 contains a discontinuity, we have $\beta_2 = O(1)$. $u^{*(0)}$ is a second order approximation since

$$\alpha_0 = \frac{1}{\Delta x^2},$$
$$\alpha_1 = \frac{1}{\Delta x^3}\left(\frac{1}{C_1} + O(\Delta x)\right),$$
$$\alpha_2 = O(1);$$

$$\omega_0 = \frac{\frac{1}{\Delta x^2}}{\frac{1}{\Delta x^2} + \frac{1}{\Delta x^3}\left(\frac{1}{C_1} + O(\Delta x)\right) + O(1)} = C_1 \Delta x + O(\Delta x^2),$$

$$\omega_2 = \frac{O(1)}{\frac{1}{\Delta x^2} + \frac{1}{\Delta x^3}\left(\frac{1}{C_1} + O(\Delta x)\right) + O(1)} = C_1 \Delta x^3 + O(\Delta x^4).$$

If $u' = 0$ but $u'' \neq 0$, then (2.17) and (2.18) reduce to

$$\beta_1 = \frac{1}{4} u''^2 \Delta x^4 + O(\Delta x^5),$$
$$\beta_2 = \frac{4}{3} u''^2 \Delta x^4 + O(\Delta x^5).$$

The above estimates can be done in a similar fashion. We only show (2.14) here. Setting $C_2 = (u'')^4$, we readily check

$$\alpha_0 = \frac{\Delta x^2}{\Delta x^4},$$

$$\alpha_1 = \frac{\Delta x}{\frac{1}{16} C_2 \Delta x^8 + O(\Delta x^9)} = \frac{\Delta x}{\Delta x^8}\left(\frac{16}{C_2} + O(\Delta x)\right),$$

$$\alpha_2 = \frac{1 + O(\Delta x)}{\Delta x^8}\left(\frac{9}{16 C_2} + O(\Delta x)\right);$$

$$\omega_0 = \frac{\Delta x^2}{\Delta x^2 + \frac{\Delta x}{\Delta x^4}\left(\frac{16}{C_2} + O(\Delta x)\right) + \frac{1+O(\Delta x)}{\Delta x^4}\left(\frac{9}{16 C_2} + O(\Delta x)\right)} = \frac{16 C_2}{9} \Delta x^6 + O(\Delta x^7),$$

$$\omega_1 = \frac{\frac{\Delta x}{\Delta x^4}\left(\frac{16}{C_2} + O(\Delta x)\right)}{\Delta x^2 + \frac{\Delta x}{\Delta x^4}\left(\frac{16}{C_2} + O(\Delta x)\right) + \frac{1+O(\Delta x)}{\Delta x^4}\left(\frac{9}{16 C_2} + O(\Delta x)\right)} = \frac{256}{9} \Delta x + O(\Delta x^2).$$

2.3. Linear stability

In this section, we prove the stability of our numerical boundary conditions for linear wave equations according to the theory of Gustafsson et al. [5] (henceforth GKS). For semi-discrete schemes, the GKS stability is studied in Strikwerda [18]. We use Strikwerda's procedure as a validation tool here. We consider quarter plane problems

$$\begin{cases} u_t + u_x = 0 & x \in (0, +\infty), \quad t > 0, \\ u(0,t) = g(t) = 0 & t > 0, \\ u(x,0) = u_0(x) & x \in [0, +\infty), \end{cases} \tag{2.19}$$

and

$$\begin{cases} u_t - u_x = 0 & x \in (0, +\infty), \quad t > 0, \\ u(x,0) = u_0(x) & x \in [0, +\infty), \end{cases} \tag{2.20}$$

for inflow boundary conditions and outflow boundary conditions, respectively. The stability of each individual case insures the stability of our method on a bounded domain. We assume the dissipative semi-discrete interior scheme has the form

$$\frac{du_j}{dt} = \frac{1}{\Delta x} \sum_{i=-r}^{p} a_i u_{j+i}, \quad j \geqslant 0. \tag{2.21}$$

For example, the fifth order WENO scheme with linear weights applied to (2.19) reads

$$\frac{du_j}{dt} = \frac{1}{\Delta x}\left(\frac{1}{30}u_{j-3} - \frac{1}{4}u_{j-2} + u_{j-1} - \frac{1}{3}u_j - \frac{1}{2}u_{j+1} + \frac{1}{20}u_{j+2}\right), \quad j \geqslant 0. \tag{2.22}$$

To solve (2.19), we use the inverse Lax-Wendroff type procedure to impose the inflow boundary condition by

$$u_v = 0, \quad v = -1, \ldots, -r. \tag{2.23}$$

An eigenvalue problem is formed by assuming a solution of the form $u(x_j,t) = u_j e^{\phi t}$. The resolvent equation for (2.21) is

$$\Delta x \phi u_j = \sum_{i=-r}^{p} a_i u_{j+i}.$$

To solve this difference equation, we set $u_j = \kappa^j u_0$ and obtain the characteristic equation

$$\Delta x \phi = \sum_{i=-r}^{p} a_i \kappa^i. \tag{2.24}$$

For Re $(\phi) > 0$, we assume κ_i, $i = 1, \ldots, q$, are all the distinct roots of (2.24) which satisfies $|\kappa_i| < 1$, each with multiplicity γ_i. A general candidate eigensolution has the form

$$u_j = \sum_{i=1}^{q} \sum_{k=1}^{\gamma_i} \alpha_{i,k} j^{k-1} (\kappa_i(\phi))^j.$$

The boundary condition (2.23) yields a linear system of equations for $\alpha_{i,k}$

$$\sum_{i=1}^{q} \sum_{k=1}^{\gamma_i} v^{k-1} \kappa_i^v \alpha_{i,k} = 0, \quad v = -1, \ldots, -r. \tag{2.25}$$

Notice that (2.25) is a square system since

$$\sum_{i=1}^{q} \gamma_i = r.$$

Denoting the coefficient matrix by $\boldsymbol{E}$, it can be shown, by elementary column operations, that $\boldsymbol{E}$ reduces to a generalized Vandermonde matrix. Thus

$$\det \boldsymbol{E} = C\left[\prod_{i=1}^{q} \kappa_i^{\rho_i}\right]\left[\prod_{1 \leqslant i < j \leqslant q} (\kappa_i - \kappa_j)^{\gamma_i \gamma_j}\right],$$

where C is a constant and ρ_i are integers. Since κ_i are distinct, $\det \boldsymbol{E} \neq 0$. Therefore, there are no nontrivial eigensolutions and our semi-discrete scheme (2.21) with boundary condition (2.23) is stable.

For the outflow boundary condition in (2.20), we can show that the semi-discrete problem (2.21) with the extrapolation (2.10) is stable for all s. The proof is essentially covered in [3] where a fully discrete scheme is analyzed.

We finally remark that the time step restriction of solving the system of ODEs (2.2) with our boundary treatment is not more severe than the pure initial value problem according to our computational experience. The standard CFL conditions determined by the interior schemes are used in the numerical examples in Section 3.

2.4. One-dimensional systems

We consider 1D compressible Euler equations

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x = 0, \quad x \in (-1,1),\ t > 0,$$

where the conservative variables

$$\boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}$$

and the flux

$$\boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} U_2 \\ (\gamma-1)U_3 + \frac{3-\gamma}{2}\frac{U_2^2}{U_1} \\ \left(\gamma U_3 - \frac{\gamma-1}{2}\frac{U_2^2}{U_1}\right)\frac{U_2}{U_1} \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix},$$

with appropriate boundary conditions and initial conditions. As is customary, ρ, u, p and E describe the density, velocity, pressure and total energy, respectively. The equation of state has the form

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

where $\gamma = 1.4$ for air at ordinary temperatures. We consider the right boundary $x = 1$. The left boundary can be treated in a similar fashion.

To decide the inflow and outflow boundary conditions at time t, we need a local characteristic decomposition of the PDEs. At the boundary, we denote the Jacobian matrix of the flux by

$$\boldsymbol{A}_\perp(\boldsymbol{U}_b) = \left.\frac{\partial \boldsymbol{F}(\boldsymbol{U})}{\partial \boldsymbol{U}}\right|_{\boldsymbol{U}=\boldsymbol{U}_b},$$

where $\boldsymbol{U}_b = \boldsymbol{U}(1,t)$. $\boldsymbol{A}_\perp(\boldsymbol{U}_b)$ has three eigenvalues $\lambda_1(\boldsymbol{U}_b) = u_b - c_b$, $\lambda_2(\boldsymbol{U}_b) = u_b$, $\lambda_3(\boldsymbol{U}_b) = u_b + c_b$ and a complete set of left eigenvectors $\boldsymbol{l}_1(\boldsymbol{U}_b)$, $\boldsymbol{l}_2(\boldsymbol{U}_b)$, $\boldsymbol{l}_3(\boldsymbol{U}_b)$ which forms a matrix

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

As in the scalar case, we assume at present that all the eigenvalues are bounded away from zero. The number of boundary conditions depends on the number of nonpositive eigenvalues. For simplicity, we assume that $U_m(1,t) = g_m(t)$, $1 \leqslant m \leqslant q$ is prescribed at the boundary if $\lambda_m(\boldsymbol{U}_b) < 0$, $1 \leqslant m \leqslant q$. Indeed, this is equivalent to prescribing the ingoing characteristic variable $\boldsymbol{V}^{\mathrm{I}}$ as a function of the outgoing characteristic variable $\boldsymbol{V}^{\mathrm{II}}$, where $\boldsymbol{V}^{\mathrm{I}}$ is the first q components of $\boldsymbol{V} = \boldsymbol{L}(\boldsymbol{U}_b)\boldsymbol{U}$ and $\boldsymbol{V}^{\mathrm{II}}$ is the last $3-q$ components of $\boldsymbol{V}$.

With the local characteristic decomposition, our problem reduces to three scalar conservation laws in terms of the characteristic variable $\boldsymbol{V}$. In our numerical scheme, we use the inverse Lax-Wendroff procedure for inflow boundary conditions of $\boldsymbol{V}^{\mathrm{I}}$ and extrapolation for outflow boundary conditions of $\boldsymbol{V}^{\mathrm{II}}$. To illustrate how this works, we assume $\lambda_1(\boldsymbol{U}_b) \leqslant \lambda_2(\boldsymbol{U}_b) < 0$ and $\lambda_3(\boldsymbol{U}_b) > 0$ at the boundary. Thus an appropriate set of boundary conditions is

$$U_1(1,t) = g_1(t),$$
$$U_2(1,t) = g_2(t),$$

for $t > 0$.

We still use the uniform mesh described in (2.6). At time level t_n, we assume $\boldsymbol{U}_j$, $j = 0,\ldots,N$, have been updated by the interior scheme. The values of ghost points are approximated by a $(s-1)$th order Taylor expansion

$$(U_m)_j = \sum_{k=0}^{s-1} \frac{(x_j - 1)^k}{k!} U_m^{*(k)}, \quad m = 1,2,3, \quad j = N+1, N+2, N+3, \tag{2.26}$$

where $U_m^{*(k)}$ is a $(s-k)$th order approximation of the spatial derivatives $\frac{\partial^k U_m}{\partial x^k}(1,t_n)$. In the local characteristic decomposition, we replace $\boldsymbol{U}_b$ by $\boldsymbol{U}_N$ since $\boldsymbol{U}(1,t_n)$ is not always prescribed. We define the outgoing local characteristic variable V_3 at grid points near the boundary by

$$(V_3)_j = \boldsymbol{l}_3(\boldsymbol{U}_N)\boldsymbol{U}_j, \quad j = N-4,\ldots,N. \tag{2.27}$$

We extrapolate $(V_3)_j$ to the boundary either with (2.12) if the solution is smooth near the boundary or with the WENO type extrapolation (2.13) if a shock is close to the boundary. We denote the extrapolated kth order derivatives of V_3 at the boundary by $V_3^{*(k)}$, $k = 0,\ldots,s-1$. Obviously, we impose $U_1^{*(0)} = g_1(t_n)$ and $U_2^{*(0)} = g_2(t_n)$. $U_3^{*(0)}$ is obtained by

$$U_3^{*(0)} = \frac{1}{l_{3,3}(\boldsymbol{U}_N)}\left[V_3^{*(0)} - l_{3,1}(\boldsymbol{U}_N)U_1^{*(0)} - l_{3,2}(\boldsymbol{U}_N)U_2^{*(0)}\right].$$

Next we try to find the spatial derivatives $U_m^{*(1)}$ with the inverse Lax-Wendroff procedure for U_1 and U_2, together with the extrapolation of V_3. The first two equations of the Euler system and the extrapolated value $V_3^{*(1)}$ give us

$$\begin{aligned}
&U_2^{*(1)} = -g_1'(t_n),\\
&(\gamma-1)U_3^{*(1)} + \frac{3-\gamma}{2}\left[\frac{2U_2^{*(0)}}{U_1^{*(0)}}U_2^{*(1)} - \left(\frac{U_2^{*(0)}}{U_1^{*(0)}}\right)^2 U_1^{*(1)}\right] = -g_2'(t_n),\\
&l_{3,1}(\boldsymbol{U}_N)U_1^{*(1)} + l_{3,2}(\boldsymbol{U}_N)U_2^{*(1)} + l_{3,3}(\boldsymbol{U}_N)U_3^{*(1)} = V_3^{*(1)}.
\end{aligned} \tag{2.28}$$

Solving this linear system, we obtain first order derivatives $U_m^{*(1)}$. We remark that the condition number of the coefficient matrix in (2.28) is small with the assumption that all the eigenvalues are bounded away from zero. Repeatedly using the Euler equations, we are able to form a linear system with the spatial derivatives $U_m^{*(k)}$ as the unknowns and the right-hand side depending on derivatives of g_1, g_2, extrapolated derivatives of V_3 and lower order spatial derivatives $U_m^{*(l)}$, $l = 0,\ldots,k-1$. The coefficient matrix always depends only on $U_m^{*(0)}$. To save space, we only provide the linear system in the case of $k = 2$ as follows:

$$\begin{aligned}
&\frac{\gamma-3}{2}\left(\frac{U_2^{*(0)}}{U_1^{*(0)}}\right)^2 U_1^{*(2)} - \frac{U_2^{*(0)}}{U_1^{*(0)}}(\gamma-3)U_2^{*(2)} + (\gamma-1)U_3^{*(2)}\\
&\quad = g_1''(t_n) + \frac{(\gamma-3)\left(U_1^{*(0)}U_2^{*(1)} - U_2^{*(0)}U_1^{*(1)}\right)^2}{\left(U_1^{*(0)}\right)^3},\\
&\quad\left[\frac{(-7+2\gamma+\gamma^2)}{2}\left(\frac{U_2^{*(0)}}{U_1^{*(0)}}\right)^3 - \gamma(\gamma-1)\frac{U_2^{*(0)}U_3^{*(0)}}{\left(U_1^{*(0)}\right)^2}\right]U_1^{*(2)}\\
&\qquad + \left[-\frac{(-12+5\gamma+\gamma^2)}{2}\left(\frac{U_2^{*(0)}}{U_1^{*(0)}}\right)^2 + \gamma(\gamma-1)\frac{U_3^{*(0)}}{U_1^{*(0)}}\right]U_2^{*(2)} + 3(\gamma-1)\frac{U_2^{*(0)}}{U_1^{*(0)}}U_3^{*(2)}\\
&= g_2''(t_n) + I_1, l_{3,1}(\boldsymbol{U}_N)U_1^{*(2)} + l_{3,2}(\boldsymbol{U}_N)U_2^{*(2)} + l_{3,3}(\boldsymbol{U}_N)U_3^{*(2)} = V_3^{*(2)},
\end{aligned}$$

where

$$\begin{aligned}
I_1 = \frac{1}{2\left(U_1^{*(0)}\right)^4}\Big\{&3(-7+2\gamma+\gamma^2)\left(U_2^{*(0)}\right)^3\left(U_1^{*(1)}\right)^2 - U_1^{*(0)}\left(U_2^{*(0)}\right)^2(-45+16\gamma+5\gamma^2)U_1^{*(1)}U_2^{*(1)} + 2(\gamma-1)\left(U_1^{*(0)}\right)^2\\
&\times\left[-(3+\gamma)U_1^{*(0)}U_2^{*(1)}U_3^{*(1)} + 2\gamma U_3^{*(0)}U_1^{*(1)}U_2^{*(1)}\right] + 2U_1^{*(0)}U_2^{*(0)}\left[-2(\gamma-1)\gamma U_3^{*(0)}\left(U_1^{*(1)}\right)^2 + U_1^{*(0)}I_2\right]\Big\},
\end{aligned}$$

$$I_2 = (-12+5\gamma+\gamma^2)\left(U_2^{*(1)}\right)^2 + (\gamma-1)(3+\gamma)U_1^{*(1)}U_3^{*(1)}.$$

Before we summarize our algorithm, we turn to the issue of zero eigenvalues. For simplicity, we assume $\lambda_2(\boldsymbol{U}_N)$ is close to zero, or the Mach number satisfies $-\beta \leqslant M_N = u_N/c_N \leqslant \beta$, where β is a small positive number, say $\beta = 0.01$. In this situation, the local characteristic variable V_2 can either go into the boundary or go out of the boundary. On the one hand, if we assume an ingoing V_2, we still have the linear system (2.28), which is however ill-conditioned. The condition number will be larger and larger as M_N gets close to zero. To make the system well-conditioned, we *add to* (2.28) the equation of extrapolation of V_2

$$l_{2,1}(\boldsymbol{U}_N)U_1^{*(1)} + l_{2,2}(\boldsymbol{U}_N)U_2^{*(1)} + l_{2,3}(\boldsymbol{U}_N)U_3^{*(1)} = V_2^{*(1)}. \tag{2.29}$$

(2.28) and (2.29) form a system of four equations with three unknowns. We consider it as a linear least squares problem. The condition number of the 4×3 matrix is small. On the other hand, if only one boundary condition is prescribed, such as the case of solid wall boundary condition $u = 0$, we have to assume an outgoing V_2. Now the second equation of (2.28) is *replaced by* (2.29) and the resulting system is well-conditioned as well.

We summarize our algorithm for imposing the values of $\boldsymbol{U}_j$ at ghost points near the right boundary $x = 1$ as follows, assuming that $\boldsymbol{U}_j$, $j = 0,\ldots,N$, have been updated at time level t_n.

1. Compute the eigenvalues $\lambda_m(\boldsymbol{U}_N)$ and left eigenvectors $\boldsymbol{l}_m(\boldsymbol{U}_N)$ of the Jacobian matrix $\boldsymbol{A}_\perp(\boldsymbol{U}_N)$ for $m = 1, 2, 3$. Decide the prescribed inflow boundary conditions $g_m(t)$ according to the signs of $\lambda_m(\boldsymbol{U}_N)$.
2. Form the outgoing characteristic variables $(V_m)_j$, $j = N-4,\ldots,N$, as in (2.27). Extrapolate $(V_m)_j$ to the boundary to obtain $V_m^{*(k)}$, $k = 0,\ldots,s-1$, with Lagrange extrapolation (2.12) or the WENO type extrapolation (2.13).
3. Solve for $U_m^{*(0)}$, $m = 1,2,3$, with the prescribed boundary conditions and extrapolated values $V_m^{*(0)}$.
4. For $k = 1,\ldots,s-1$, use the inverse Lax-Wendroff type procedure to write the kth order derivatives of $g_m(t)$ as a linear combination of kth order spatial derivatives and other terms. Together with the extrapolation equations, form a linear system (such as (2.28)) or a linear least squares problem (such as (2.28) and (2.29)) for $U_m^{*(k)}$, $m = 1,2,3$. Solve for $U_m^{*(k)}$, $m = 1,2,3$.
5. Impose the values of the ghost points by the Taylor expansion (2.26).

2.5. Two-dimensional case

The approach can be easily generalized to two-dimensional problems (2.1). We consider 2D compressible Euler equations

$$\boldsymbol{U}_t + \boldsymbol{F}(\boldsymbol{U})_x + \boldsymbol{G}(\boldsymbol{U})_y = 0, \quad (x,y) \in \Omega,\ t > 0, \tag{2.30}$$

where

$$\boldsymbol{U} = \begin{pmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix},$$

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

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

with appropriate boundary conditions and initial conditions. ρ, u, v, p and E describe the density, x-velocity, y-velocity, pressure and total energy, respectively. The equation of state has the form

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

where $\gamma = 1.4$ for air at ordinary temperatures.

We assume the values of the grid points inside domain Ω have been updated by the interior scheme. To define the value of $\boldsymbol{U}_{i,j}$ at a ghost point $P = (x_i, y_j)$, we find a point $P_0 = (x_0, y_0)$ on the boundary $\partial\Omega$ so that the outward normal $\boldsymbol{n} = (n_1, n_2)$ to $\partial\Omega$ at P_0 goes through P. We set up a local coordinate system at P_0 by

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \boldsymbol{T} \begin{pmatrix} x \\ y \end{pmatrix}, \tag{2.31}$$

where θ is the angle between the normal $\boldsymbol{n}$ and the x-axis and $\boldsymbol{T}$ is a rotation matrix. Notice that the $\hat{x}$-axis points in the normal direction to $\partial\Omega$ at P_0 and the $\hat{y}$-axis points in the tangential direction to $\partial\Omega$ at P_0. In this local coordinate system, the Euler system (2.30) is

$$\widehat{\boldsymbol{U}}_t + \boldsymbol{F}(\widehat{\boldsymbol{U}})_x + \boldsymbol{G}(\widehat{\boldsymbol{U}})_y = 0, \tag{2.32}$$

where

$$\widehat{\boldsymbol{U}} = \begin{pmatrix} \widehat{U}_1 \\ \widehat{U}_2 \\ \widehat{U}_3 \\ \widehat{U}_4 \end{pmatrix} = \begin{pmatrix} \rho \\ \rho\hat{u} \\ \rho\hat{v} \\ E \end{pmatrix},$$

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

At the boundary, we have the Jacobian matrix of the normal flux

$$\boldsymbol{A}_\perp(\widehat{\boldsymbol{U}}_b)=\left.\frac{\partial \boldsymbol{F}(\widehat{\boldsymbol{U}})}{\partial \widehat{\boldsymbol{U}}}\right|_{\widehat{\boldsymbol{U}}=\widehat{\boldsymbol{U}}_b},$$

where $\widehat{\boldsymbol{U}}_b=\widehat{\boldsymbol{U}}(x_0,y_0,t)$. $\boldsymbol{A}_\perp(\widehat{\boldsymbol{U}}_b)$ has four eigenvalues $\lambda_1(\widehat{\boldsymbol{U}}_b)=\hat{u}_b-c_b,\ \lambda_2(\widehat{\boldsymbol{U}}_b)=\lambda_3(\widehat{\boldsymbol{U}}_b)=\hat{u}_b,\ \lambda_4(\widehat{\boldsymbol{U}}_b)=\hat{u}_b+c_b$ and a complete set of left eigenvectors $\boldsymbol{l}_1(\widehat{\boldsymbol{U}}_b),\ \boldsymbol{l}_2(\widehat{\boldsymbol{U}}_b),\boldsymbol{l}_3(\widehat{\boldsymbol{U}}_b),\ \boldsymbol{l}_4(\widehat{\boldsymbol{U}}_b)$ which forms a matrix

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

Now the number of prescribed boundary conditions depends on the signs of $\lambda_m(\widehat{\boldsymbol{U}}_b),\ m=1,\dots,4$. For simplicity, we assume that $\lambda_m(\widehat{\boldsymbol{U}}_b)\leqslant-\alpha<0,\ m=1,2,3$ and $\lambda_4(\widehat{\boldsymbol{U}}_b)\geqslant\alpha>0$. The boundary conditions are

$$\begin{aligned}&\widehat{U}_1(x_0,y_0,t)=g_1(t),\\ &\widehat{U}_2(x_0,y_0,t)=g_2(t),\\ &\widehat{U}_3(x_0,y_0,t)=g_3(t),\end{aligned}$$

for $t>0$.

The value of $\widehat{\boldsymbol{U}}_{ij}$ at ghost point P is approximated by a $(s-1)$th order Taylor expansion

$$(\widehat{U}_m)_{i,j}=\sum_{k=0}^{s-1}\frac{d^k}{k!}\widehat{U}_m^{*(k)},\quad m=1,\dots,4, \tag{2.33}$$

where d is the $\hat{x}$-coordinate of P and $\widehat{U}_m^{*(k)}$ is a $(s-k)$th order approximation of the normal derivatives $\frac{\partial^k\widehat{U}_m}{\partial\hat{x}^k}(x_0,y_0,t_n)$. In the local characteristic decomposition, we replace $\widehat{\boldsymbol{U}}_b$ by $\widehat{\boldsymbol{U}}_0$ which is the value of a grid point nearest to P_0 among all the grid points inside Ω. We define the outgoing local characteristic variable V_4 at grid points near P_0 by

$$(V_4)_{\mu,\nu}=\boldsymbol{l}_4(\widehat{\boldsymbol{U}}_0)\widehat{\boldsymbol{U}}_{\mu,\nu},\quad (x_\mu,y_\nu)\in\mathcal{E}_{ij}, \tag{2.34}$$

where $\mathcal{E}_{ij}$ is a set of grid points inside Ω which are used to construct an extrapolating polynomial. The construction of $\mathcal{E}_{ij}$ will be discussed in Section 2.6. We extrapolate $(V_4)_{\mu,\nu}$ to the boundary and denote the extrapolated $\hat{x}$-derivatives of V_4 at the boundary by $V_4^{*(k)},\ k=0,\dots,s-1$. Obviously, we impose $\widehat{U}_1^{*(0)}=g_1(t_n),\ \widehat{U}_2^{*(0)}=g_2(t_n)$ and $\widehat{U}_3^{*(0)}=g_3(t_n)$. $\widehat{U}_4^{*(0)}$ is obtained by

$$\widehat{U}_4^{*(0)}=\frac{1}{l_{4,4}(\widehat{\boldsymbol{U}}_0)}\left[V_4^{*(0)}-l_{4,1}(\widehat{\boldsymbol{U}}_0)\widehat{U}_1^{*(0)}-l_{4,2}(\widehat{\boldsymbol{U}}_0)\widehat{U}_2^{*(0)}-l_{4,3}(\widehat{\boldsymbol{U}}_0)\widehat{U}_3^{*(0)}\right].$$

Next we try to find the $\hat{x}$-derivatives $\widehat{U}_m^{*(1)}$ with the inverse Lax-Wendroff procedure for $\widehat{U}_1,\ \widehat{U}_2,\ \widehat{U}_3$, together with the extrapolation of V_4. The first three equations of (2.32) and the extrapolated value $V_4^{*(1)}$ give us

$$\boldsymbol{A}\widehat{\boldsymbol{U}}^{*(1)}=\boldsymbol{b}, \tag{2.35}$$

where

$$\boldsymbol{A}=\begin{pmatrix}0 & 1 & 0 & 0\\ \frac{\gamma-3}{2}\left(\frac{U_2^{*(0)}}{U_1^{*(0)}}\right)^2+\frac{\gamma-1}{2}\left(\frac{U_3^{*(0)}}{U_1^{*(0)}}\right)^2 & (3-\gamma)\frac{U_2^{*(0)}}{U_1^{*(0)}} & (1-\gamma)\frac{U_3^{*(0)}}{U_1^{*(0)}} & (\gamma-1)\\ -\frac{U_2^{*(0)}}{U_1^{*(0)}}\frac{U_3^{*(0)}}{U_1^{*(0)}} & \frac{U_3^{*(0)}}{U_1^{*(0)}} & \frac{U_2^{*(0)}}{U_1^{*(0)}} & 0\\ l_{4,1}(\widehat{\boldsymbol{U}}_0) & l_{4,2}(\widehat{\boldsymbol{U}}_0) & l_{4,3}(\widehat{\boldsymbol{U}}_0) & l_{4,4}(\widehat{\boldsymbol{U}}_0)\end{pmatrix},$$

$$\widehat{\boldsymbol{U}}^{*(1)}=\begin{pmatrix}\widehat{U}_1^{*(1)}\\ \widehat{U}_2^{*(1)}\\ \widehat{U}_3^{*(1)}\\ \widehat{U}_4^{*(1)}\end{pmatrix}$$

and

$$\boldsymbol{b} = \begin{pmatrix} -g_1'(t_n) - \frac{\partial}{\partial \tilde{y}}\left(\widehat{U}_3^{*(0)}\right) \\ -g_2'(t_n) - \frac{\partial}{\partial \tilde{y}}\left(\frac{\widehat{U}_2^{*(0)}\widehat{U}_3^{*(0)}}{\widehat{U}_1^{*(0)}}\right) \\ -g_3'(t_n) - \frac{\partial}{\partial \tilde{y}}\left[(\gamma-1)\widehat{U}_4^{*(0)} - \frac{\gamma-1}{2}\frac{\left(\widehat{U}_2^{*(0)}\right)^2}{\widehat{U}_1^{*(0)}} + \frac{3-\gamma}{2}\frac{\left(\widehat{U}_3^{*(0)}\right)^2}{\widehat{U}_1^{*(0)}}\right] \\ V_4^{*(1)} \end{pmatrix}.$$

Solving this 4×4 linear system, we obtain first order normal derivatives $\widehat{U}_m^{*(1)}$. Again, the condition number of $\boldsymbol{A}$ in (2.35) is small with the assumption that all the eigenvalues are bounded away from zero. Repeatedly using the Euler equations, we are able to form a linear system with the normal derivatives $\widehat{U}_m^{*(k)}$ as the unknowns and the right-hand side depending on derivatives of g_1, g_2, g_3, tangential derivatives, extrapolated normal derivatives of V_4 and lower order normal derivatives $\widehat{U}_m^{*(l)}$, $l = 0, \ldots, k-1$. The coefficient matrix depends only on $\widehat{U}_m^{*(0)}$.

To compute the tangential derivatives in $\boldsymbol{b}$, we use either analytical expressions if available or numerical differentiation. The latter is always possible since we have obtained the values of $\widehat{\boldsymbol{U}}_m^{*(0)}$ along the boundary. For robustness, a least squares polynomial of suitable degree is used if the solution is smooth near the boundary, or WENO type differentiation is used otherwise.

We now summarize our algorithm for imposing the values of $\boldsymbol{U}_{i,j}$ at ghost points as follows, assuming the values of the grid points inside Ω have been updated at time level t_n.

1. For each ghost point (x_i, y_j), we do the following three steps:
   - Decide the local coordinate system (2.31). Compute the eigenvalues $\lambda_m(\widehat{\boldsymbol{U}}_0)$ and left eigenvectors $\boldsymbol{l}_m(\widehat{\boldsymbol{U}}_0)$ of the Jacobian matrix $\boldsymbol{A}_\perp(\widehat{\boldsymbol{U}}_0)$ for $m = 1,\ldots,4$. Decide the prescribed inflow boundary conditions $g_m(t)$ according to the signs of $\lambda_m(\widehat{\boldsymbol{U}}_0)$.
   - Form the outgoing characteristic variables $(V_m)_{\mu,\nu}, (x_\mu, y_\nu) \in \mathcal{E}_{i,j}$ as in (2.34). Extrapolate $(V_m)_{\mu,\nu}$ to the boundary to obtain $V_m^{*(k)}$, $k = 0, \ldots, s-1$, with a Lagrange polynomial or a least squares polynomial if the solution is smooth near the boundary, or with the WENO type extrapolation otherwise. Details of the two-dimensional extrapolation will be discussed in Section 2.6.
   - Solve for $\widehat{U}_m^{*(0)}$, $m = 1, \ldots, 4$, with the prescribed boundary conditions and extrapolated values $V_m^{*(0)}$.
2. For $k = 1,\ldots,s-1$, we do the following calculations. For each ghost point (x_i,y_j), use the inverse Lax-Wendroff type procedure to write the kth order derivatives of $g_m(t)$ as a linear combination of kth order normal derivatives plus tangential derivatives and other terms. Together with the extrapolation equations, form a linear system (such as (2.35)) or a linear least squares problem for $\widehat{U}_m^{*(k)}$, $m = 1, \ldots, 4$. Solve for $\widehat{U}_m^{*(k)}$, $m = 1, \ldots, 4$.
3. Impose the value of $\widehat{\boldsymbol{U}}_{i,j}$ by the Taylor expansion (2.33) and transform $\widehat{\boldsymbol{U}}_{i,j}$ to $\boldsymbol{U}_{i,j}$.

We remark that the change of coordinate system is used only for determining inflow boundary conditions and for applying the inverse Lax-Wendroff procedure. Our uniform Cartesian mesh remains unchanged.

2.6. Two-dimensional extrapolation

We return to the issue of two-dimensional extrapolation, which is needed in the second bullet of step 1 in our algorithm flowchart for two-dimensional problems. We assume $u_{i,j}$, $(x_i,y_j) \in \Omega$, are given. We aim to first construct a stencil $\mathcal{E} \subset \Omega$ for extrapolation and then obtain a $(s-k)$th order approximation of $\frac{\partial^k u}{\partial \hat{x}^k}|_{(x,y)=(x_0,y_0)}$, which is denoted by $u^{*(k)}$, $k = 0,\ldots,s-1$. We assume $s = 3$, since we use this order of extrapolation in most of the numerical examples.

The choice of stencil $\mathcal{E}$ is sketched in Fig. 2.1. $\mathcal{E}$ contains three one-dimensional substencils $\mathcal{S}_l$, $l = 0, 1, 2$. Suppose the normal $\boldsymbol{n}$ (or $\hat{x}$-axis) intersects the grid line $y = y_{n+l}$, $l = 0, 1, 2$, at a point P_l^*. We identify the grid point on $y = y_{n+l}$ which is nearest to P_l^* by (x_m^l, y_{n+l}). Then we set $\mathcal{S}_l = \{(x_{m-1}^l, y_{n+l}), (x_m^l, y_{n+l}), (x_{m+1}^l, y_{n+l})\}$, $l = 0, 1, 2$ and $\mathcal{E} = \bigcup_{l=0}^2 \mathcal{S}_l$. Notice that we might need to shift $\mathcal{S}_l$ to the left or to the right so that $\mathcal{S}_l$ lies in Ω.

Once the stencil $\mathcal{E}$ is chosen, we can easily construct a Lagrange polynomial in Q^2

$$p_2(x,y) = \sum_{m=0}^{2}\sum_{l=0}^{2} a_{lm} x^l y^m$$

satisfying

$$p_2(x_i,y_j) = u_{ij}, \quad (x_i,y_j) \in \mathcal{E}. \tag{2.36}$$

$u^{*(k)}$ is then obtained by

$$u^{*(k)} = \frac{\partial^k}{\partial \hat{x}^k} p_2(x,y)\Big|_{(x,y)=(x_0,y_0)}. \tag{2.37}$$

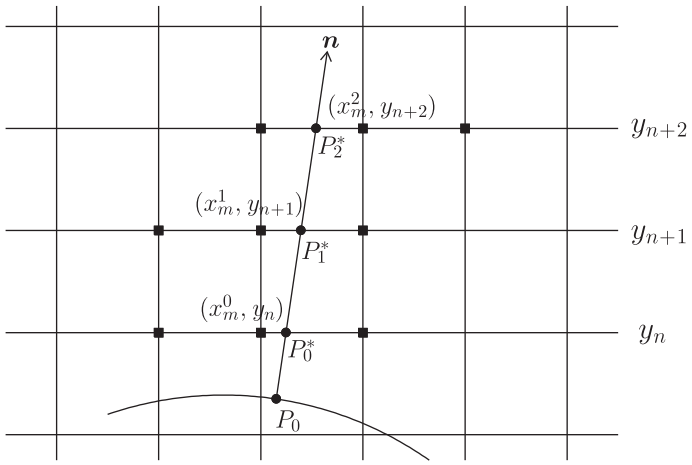


Fig. 2.1. The choice of stencil $\mathcal{E}$ (square points) for two-dimensional extrapolation.

We can also develop WENO type extrapolation, which is useful if a shock is close to the boundary. Suppose

$$p_r(x,y) = \sum_{m=0}^{r}\sum_{l=0}^{r} a_{lm}x^l y^m, \quad r = 0,1,2,$$

is constructed on S_r^2, where $S_0^2 \subset S_1^2 \subset S_2^2 = \mathcal{E}$ are two-dimensional substencils containing $(r+1)^2$ points. We seek WENO type extrapolation of the form

$$u^{*(k)} = \sum_{r=0}^{2} \omega_r \left.\frac{\partial^k p_r(x,y)}{\partial \hat{x}^k}\right|_{(x,y)=(x_0,y_0)}, \tag{2.38}$$

where ω_r are nonlinear weights. The nonlinear weights ω_r are chosen to be

$$\omega_r = \frac{\alpha_r}{\sum_{s=0}^{2}\alpha_s},$$

with

$$\alpha_r = \frac{d_r}{(\epsilon+\beta_r)^2},$$

where $\epsilon = 10^{-6}$, $d_0 = \Delta x^2 + \Delta y^2$, $d_1 = \sqrt{\Delta x^2 + \Delta y^2}$, $d_2 = 1 - d_0 - d_1$. β_r are determined by

$$\beta_0 = \Delta x^2 + \Delta y^2,$$
$$\beta_r = \sum_{1\leqslant|\alpha|\leqslant r}\int_K |K|^{|\alpha|-1}\left(D^\alpha p_r(x,y)\right)^2 dx\,dy, \quad r = 1,2,$$

where α is a multi-index and $K = [x_0 - \Delta x/2, x_0 + \Delta x/2] \times [y_0 - \Delta y/2, y_0 + \Delta y/2]$. We can show that β_r, $r = 1$ or 2, is small if a discontinuity appears in S_r^2 and

$$\begin{aligned}\omega_0 &= O(\Delta x^2 + \Delta y^2),\\ \omega_1 &= O(\sqrt{\Delta x^2 + \Delta y^2}),\\ \omega_2 &= 1 - \omega_0 - \omega_1,\end{aligned} \tag{2.39}$$

if the values of $u_{i,j}$ are smooth. We omit the proof here since the arguments are very similar to those in Section 2.2.

In the case of subsonic outflow boundary conditions, the high order Lagrange extrapolation (2.37) seems to make the whole scheme mildly unstable. A remedy is to include more points in $\mathcal{E}$ so that (2.36) is satisfied in the sense of least squares. We use this least squares extrapolation in the vortex evolution problem in Section 3.2, where $\mathcal{E}$ is taken as

$$\mathcal{E} = \left\{(x_\mu, y_\nu) \in \Omega : \sqrt{(x_\mu - x_0)^2 + (y_\nu - y_0)^2} < 5\Delta x\right\}.$$

It seems that the WENO type extrapolation (2.38) does not have this issue when used in the presence of shocks near the boundary. This might be due to the fact that the WENO type extrapolation reduces to a low order method if the solution becomes unstable, which helps stabilizing the scheme.

3. Numerical examples

3.1. One-dimensional examples

Example 1. We start with the wave equation

$$\begin{cases} u_t + u_x = 0 & x \in (-1,1), \quad t > 0, \\ u(x,0) = 0.25 + 0.5\sin(\pi x) & x \in [-1,1], \\ u(-1,t) = g(t) & t > 0. \end{cases} \tag{3.1}$$

The left boundary $x = -1$ is an inflow boundary, where a boundary condition is prescribed. The right boundary $x = 1$ is an outflow boundary, where no boundary condition is needed.

We first take

$$g(t) = 0.25 - 0.5\sin[\pi(1+t)] \tag{3.2}$$

so that the initial boundary value problem has a smooth exact solution

$$u(x,t) = 0.25 + 0.5\sin[\pi(x-t)].$$

We use a fourth order Taylor expansion. The errors at $t = 1$ are listed in Table 3.1. We can clearly see the desired fifth order convergence.

Next, we take $g(t)$ as

$$g(t) = \begin{cases} 0.25 & t \leqslant 1, \\ -1 & t > 1. \end{cases} \tag{3.3}$$

The exact solution is then

$$u(x,t) = \begin{cases} -1 & x < t-2, \\ 0.25 & t-2 \leqslant x < t-1, \\ 0.25 + 0.5\sin[\pi(x-t)] & x \geqslant t-1. \end{cases}$$

For $t \leqslant 1$, the exact solution has a discontinuity in its first derivative, due to the definition of $g(t)$. For $t > 1$, a discontinuity enters the computational domain from the inflow boundary. We can observe from Fig. 3.1 that both types of discontinuities are well captured by our method.

Table 3.1
Errors of the wave equation (3.1) with boundary condition (3.2). $\Delta x = 2/N$ and $t = 1$.

N	L^1 error	Order	L^∞ error	Order
40	1.74E−05		2.31E−05	
80	5.31E−07	5.03	7.42E−07	4.96
160	1.65E−08	5.01	2.21E−08	5.07
320	5.16E−10	5.00	6.66E−10	5.05
640	1.61E−11	5.00	1.88E−11	5.15

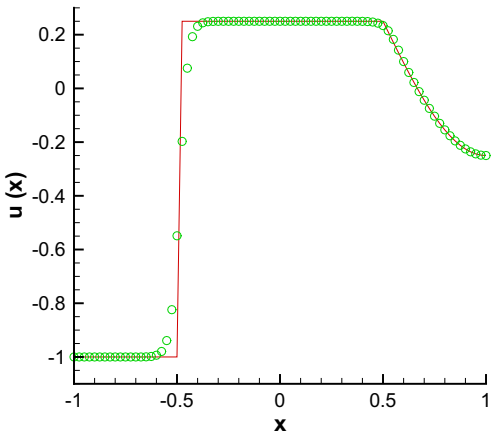


Fig. 3.1. Wave equation (3.1) with boundary condition (3.3). $\Delta x = 1/40$, $t = 1.5$. Solid line: exact solution; symbols: numerical solution.

Example 2. We next test the Burgers equation

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x = 0 & x \in (-1,1), \quad t > 0, \\ u(x,0) = 0.25 + 0.5\sin(\pi x) & x \in [-1,1], \\ u(-1,t) = g(t) & t > 0. \end{cases} \tag{3.4}$$

Here $g(t) = w(-1,t)$, where $w(x,t)$ is the exact solution of the initial value problem on $(-1,1)$ with periodic boundary conditions. For all t, the left boundary $x = -1$ is an inflow boundary and the right boundary $x = 1$ is an outflow boundary. We use a fourth order Taylor expansion. At $t = 0.3$, we have a smooth solution. The errors are listed in Table 3.2. We achieve the designed fifth order accuracy. At $t = 1.1$, a shock is fully developed in the interior of the computational domain. A shock enters the inflow boundary at $t = 8$ and moves to $x = 0$ at $t = 12$. We can see from Fig. 3.2 that the shock is well captured in both scenarios by our method.

Example 3. For one-dimensional systems, we first test the Euler equations with smooth solutions.

(a) The domain is $(-\pi,\pi)$. The initial condition is

$$\begin{aligned} &\rho(x,0) = 1 + 0.2\sin x, \\ &u(x,0) = 1, \\ &p(x,0) = 2. \end{aligned} \tag{3.5}$$

We want to impose the boundary conditions in such a way that the exact solution is simply a translation of the initial condition

$$\begin{aligned} &\rho(x,t) = 1 + 0.2\sin(x-t), \\ &u(x,t) = 1, \\ &p(x,t) = 2. \end{aligned}$$

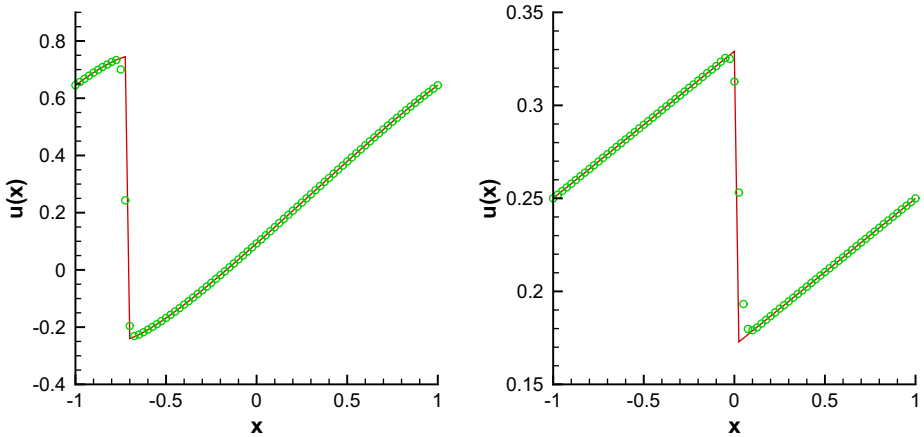
At both boundaries, we have $\lambda_1 < 0$, $\lambda_2 > 0$ and $\lambda_3 > 0$. Hence two boundary conditions are needed at $x = -\pi$, which are taken as

$$\rho(-\pi,t) = 1 + 0.2\sin t, \tag{3.6}$$

$$u(-\pi,t) = 1. \tag{3.7}$$

Table 3.2
Errors of the Burgers equation (3.4). $\Delta x = 2/N$ and $t = 0.3$.

N	L^1 error	Order	L^∞ error	Order
40	9.11E−05		3.56E−04	
80	3.10E−06	4.88	1.35E−05	4.72
160	1.31E−07	4.57	6.51E−07	4.38
320	3.97E−09	5.05	2.68E−08	4.60
640	1.02E−10	5.29	8.34E−10	5.00
1280	2.86E−12	5.15	2.62E−11	5.00



(a) $t = 1.1$ (b) $t = 12$

Fig. 3.2. Burgers equation (3.4), $\Delta x = 1/40$. Solid line: exact solution; symbols: numerical solution.

One boundary condition is needed at $x = \pi$, which is taken as

$$\rho(\pi, t) = 1 + 0.2 \sin t. \tag{3.8}$$

We use a fourth order Taylor expansion. We can observe from Table 3.3 that the designed fifth order accuracy is achieved.

(b) The domain is (0,1). The initial condition is

$$\begin{aligned} \rho(x, 0) &= a + \sin(2\pi x), \\ u(x, 0) &= 1 + 0.1 \sin(2\pi x), \\ p(x, 0) &= b, \end{aligned} \tag{3.9}$$

where a and b are real numbers. A reference solution $W(x,t)$ can be obtained by the fifth order WENO scheme with periodic boundary conditions on $(0,1)$ using an extremely refined mesh. To test our method, we assume Dirichlet boundary conditions in such a way that the exact solution of the initial boundary value problem is $W(x,t)$.

If we take $a = 1.5$, $b = 3$ in (3.9), we have $\lambda_1 < 0$, $\lambda_2 > 0$ and $\lambda_3 > 0$ at both boundaries. At the left boundary $x = 0$, we prescribe two boundary conditions

$$u_1(0, t) = W_1(0, t), \tag{3.10}$$

$$u_2(0, t) = W_2(0, t). \tag{3.11}$$

At the right boundary, we prescribe one boundary condition

$$u_1(1, t) = W_1(1, t). \tag{3.12}$$

We use a second order Taylor expansion so that a third order method is expected. For this truly nonlinear problem, we can see third order convergence in the left part of Table 3.4.

If we take $a = 2$, $b = 1.6$ in (3.9), the eigenvalues λ_m change signs at both boundaries. As a result, the number of boundary conditions varies with time. A least squares problem is solved if λ_m is close to zero. We can observe from the right part of Table 3.4 that the designed third order accuracy is again achieved.

Example 4. We then test our method for the Euler equations with shocks. We consider the interaction of two blast waves [19]. The initial data are

$$\boldsymbol{U}(x, 0) = \begin{cases} \boldsymbol{U}_L & 0 < x < 0.1, \\ \boldsymbol{U}_M & 0.1 < x < 0.9, \\ \boldsymbol{U}_R & 0.9 < x < 1, \end{cases}$$

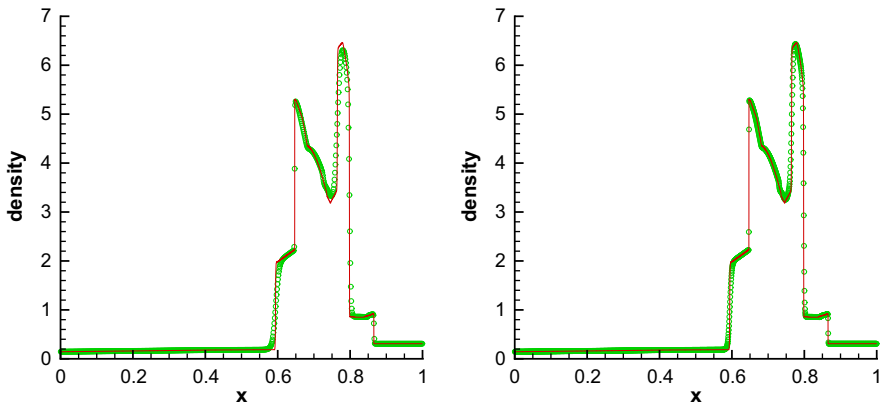
where $\rho_L = \rho_M = \rho_R = 1$, $u_L = u_M = u_R = 0$, $p_L = 10^3$, $p_M = 10^{-2}$, $p_R = 10^2$. There are solid wall boundary conditions at both $x = 0$ and $x = 1$. This problem involves multiple reflections of shocks and rarefactions off the walls. There are also multiple interactions of shocks and rarefactions with each other and with contact discontinuities. At both boundaries, we use a second order Taylor expansion with the inverse Lax-Wendroff procedure for the inflow condition $u = 0$ and the WENO type extrapolation for the outflow conditions. The density profile at $t = 0.038$ is shown in Fig. 3.3(a) with $\Delta x = 1/800$ and in Fig. 3.3(b)

Table 3.3
Density errors of the Euler equations with initial condition (3.5) and boundary conditions (3.6)–(3.8). $\Delta x = 2\pi/N$ and $t = 2$.

N	L^1 error	Order	L^∞ error	Order
40	3.23E−06		8.96E−06	
80	7.39E−08	5.45	2.00E−07	5.48
160	2.12E−09	5.12	5.91E−09	5.08
320	6.49E−11	5.03	1.66E−10	5.16
640	2.00E−12	5.02	4.91E−12	5.08

Table 3.4
Density errors of the Euler equations with initial condition (3.9). $\Delta x = 1/N$.

N	$a = 1.5$, $b = 3$, $t = 1.5$				$a = 2$, $b = 1.6$, $t = 1$			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
80	3.35E−05		1.73E−04		3.41E−06		1.29E−05	
160	3.49E−06	3.26	2.06E−05	3.07	4.72E−07	2.85	2.39E−06	2.43
320	5.36E−07	2.70	3.10E−06	2.74	6.10E−08	2.95	3.31E−07	2.85
640	7.13E−08	2.91	4.10E−07	2.92	7.69E−09	2.99	4.65E−08	2.83
1280	9.12E−09	2.97	5.24E−08	2.97	9.87E−10	2.96	6.36E−09	2.87



(a) $\Delta x = 1/800$ (b) $\Delta x = 1/1600$

Fig. 3.3. The density profiles of the blast wave problem. Solid lines: reference solution computed by the fifth order WENO scheme with $\Delta x = 1/16{,}000$, together with the reflection technique at boundaries; symbols: numerical solutions by our boundary treatment.

with $\Delta x = 1/1600$. The reference solution is computed by the fifth order WENO scheme with $\Delta x = 1/16{,}000$, together with the reflection technique at both boundaries. We can clearly see that our boundary treatment gives a satisfactory resolution.

Before we finish this section, there are some comments on the additional computational cost of the proposed high order boundary treatment. In all our one-dimensional examples, the CPU time of the boundary treatment takes up at most five percent of the entire CUP time of computation. We do not list the exact ratios since the CPU time of the boundary treatment is so small that it varies significantly for repeated runs. In other words, the additional cost of our boundary treatment is negligible compared to the cost of the entire computation for one-dimensional problems.

3.2. Two-dimensional examples

Example 5. We start our two-dimensional examples with the wave equation on a square or on a disk

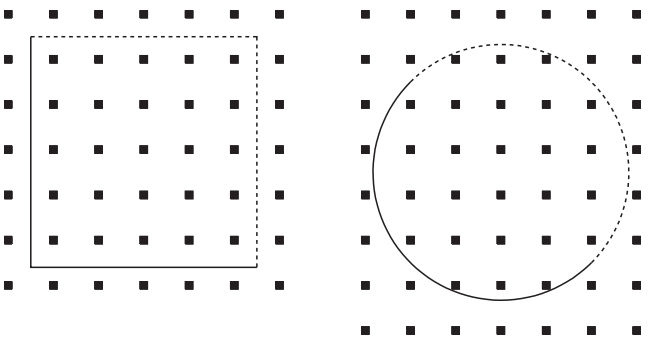
$$\begin{cases} u_t + u_x + u_y = 0 & (x,y) \in \Omega, \quad t > 0, \\ u(x,y,0) = 0.25 + 0.5\sin[\pi(x+y)] & (x,y) \in \bar{\Omega}, \\ u(x,y,t) = g(x,y,t) & (x,y) \in \Gamma, \quad t > 0, \end{cases} \tag{3.13}$$

where

$$\Omega = (-1,1) \times (-1,1),$$
$$\Gamma = \{(x,y) : x = -1 \text{ or } y = -1\},$$

or

$$\Omega = \{(x,y) : x^2 + y^2 < 0.5\},$$
$$\Gamma = \{(x,y) : x^2 + y^2 = 0.5 \text{ and } x + y \leqslant 0\}.$$



(a) square domain (b) circular domain

Fig. 3.4. Domain Ω of the 2D wave equation (3.13). Square points indicate some of the grid points. Solid lines: inflow boundary; dashed lines: outflow boundary.

Both domains are illustrated in Fig. 3.4 with a coarse mesh. Notice that the grid points are not located on the boundary in either case.

On the disk, special care must be taken when we impose the inflow boundary condition on a ghost point near the intersection of the inflow boundary and the outflow boundary, which is $(-0.5, 0.5)$ or $(0.5, -0.5)$ in our example. In this situation, the inverse Lax-Wendroff procedure involves a small number divided by a small number, which ruins the accuracy or makes the scheme blow up. There are two ways to deal with this ill-conditioned problem. The first one is using the analytical expressions of time derivatives and tangential derivatives. The other one is adding an extrapolation equation and solving a least squares problem. We use the former here because the analytical expressions are available. We will use the latter in the next example.

We take

$$g(x,y,t) = 0.25 + 0.5\sin[\pi(x+y-2t)] \tag{3.14}$$

so that we have a smooth exact solution

$$u(x,y,t) = 0.25 + 0.5\sin[\pi(x+y-2t)].$$

We use a fourth order Taylor expansion. The errors are listed in Table 3.5. We can clearly see fifth order convergence.

To study the additional computational cost, we list the ratios between the CPU time of boundary treatment and the total CPU time of computation in Table 3.6. For each problem, the finer the mesh, the lower the ratio, which results simply from the fact that the boundary is one-dimensional while the computational domain is two-dimensional. Comparing the two problems, we can see that the cost due to boundary treatment of wave equation on a square is about twice as large as that of the same equation on a disk. In fact, this is mainly due to the different ways of computing the time derivatives and tangential derivatives involved in the inverse Lax-Wendroff procedure. We use ENO type differentiation in the former problem while analytical expressions are used in the latter problem. Further numerical experiments show that the cost due to boundary treatment is negligible (under two percent) if analytical expressions are used in the former problem. We conclude here that although the inverse Lax-Wendroff procedure itself is straightforward if we have a simple PDE on a simple domain, computing the time derivatives and tangential derivatives using numerical differentiation could be rather expensive.

We next take a discontinuous boundary condition

$$g(x,y,t) = \begin{cases} 0.25 + 0.5\sin\pi(x+y-2t) & x+y-2t > -1.23, \\ 1.25 + 0.5\sin\pi(x+y-2t) & x+y-2t \leqslant -1.23. \end{cases} \tag{3.15}$$

Now we have a discontinuous exact solution. The numerical solution and exact solution along the diagonal are shown in Fig. 3.5. We can see an excellent non-oscillatory resolution.

Example 6. We next test the 2D Burgers equation

$$\begin{cases} u_t + \frac{1}{2}(u^2)_x + \frac{1}{2}(u^2)_y = 0 & (x,y) \in \Omega, \quad t > 0, \\ u(x,y,0) = 0.75 + 0.5\sin[\pi(x+y)] & (x,y) \in \bar{\Omega}, \\ u(x,y,t) = g(x,y,t) & (x,y) \in \Gamma, \quad t > 0, \end{cases} \tag{3.16}$$

where

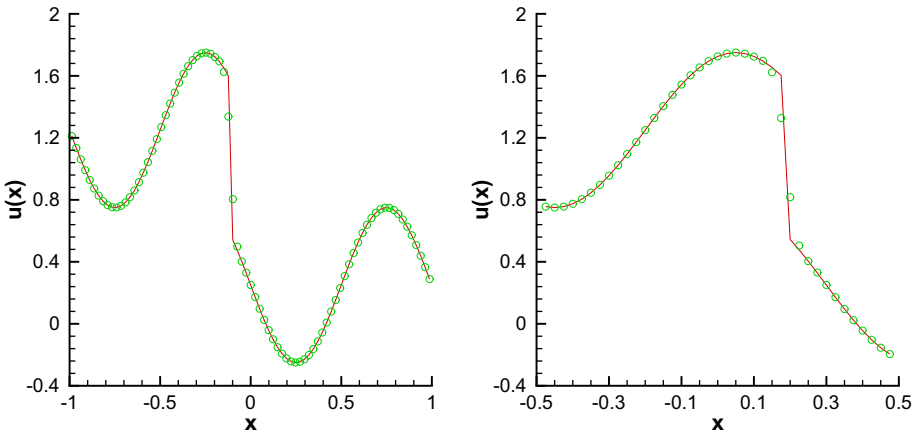
$$\Omega = (-1,1) \times (-1,1),$$
$$\Gamma = \{(x,y) : x = -1 \text{ or } y = -1\},$$

Table 3.5
Errors of the 2D wave equation (3.13) with boundary condition (3.14). $\Delta x = 2/N_x$, $\Delta y = 2/N_y$.

$N_x = N_y$	On a square, $t = 0.5$				On a disk, $t = 0.8$			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
40	8.19E−06		2.17E−05		9.64E−06		3.61E−05	
80	2.68E−07	4.93	7.11E−07	4.93	2.88E−07	5.06	9.45E−07	5.26
160	8.59E−09	4.96	2.24E−08	4.99	8.89E−09	5.02	3.01E−08	4.97
320	2.72E−10	4.98	6.77E−10	5.05	2.76E−10	5.01	9.58E−10	4.97

Table 3.6
Ratio between the CPU time of boundary treatment and the total CPU time of computation, the 2D wave equation (3.13) with boundary condition (3.14).

$N_x = N_y$	40	80	160	320
On a square	0.73	0.64	0.46	0.30
On a disk	0.42	0.33	0.20	0.12



(a) on a square, $t = 0.5$ (b) on a disk, $t = 0.8$

Fig. 3.5. 2D wave equation (3.13) with boundary condition (3.15). $\Delta x = \Delta y = 1/40$. Cut along the diagonal. Solid line: exact solution; symbols: numerical solution.

or

$$\Omega = \{(x,y) : x^2 + y^2 < 0.5\},$$
$$\Gamma = \{(x,y) : x^2 + y^2 = 0.5 \text{ and } x + y \leqslant 0\}.$$

Here $g(x,y,t) = w(x,y,t)$, where $w(x,y,t)$ is the exact solution of the initial value problem on $(-1,1) \times (-1,1)$ with periodic boundary conditions. We use a second order Taylor expansion at the inflow boundary. At the outflow boundary, we use the WENO type extrapolation, which is third order accurate if the solution is smooth. At $t = 0.15$, we have a smooth solution. The errors are listed in Table 3.7. We achieve the designed third order accuracy. At $t = 0.55$, a shock is fully developed in the interior of Ω. A shock begins entering Ω from the inflow boundary at $t = 4$ and moves to $x = 0$ at $t = 6$. We can see from Fig. 3.6 that the shock is well captured on both domains. Notice that there is a shock very close to the outflow boundary in Fig. 3.6(a). The robust WENO type extrapolation gives us a non-oscillatory numerical solution with little overshoot.

Example 7. For two-dimensional systems, we test the vortex evolution problem for the Euler equation (2.30). The mean flow is $\rho = p = u = v = 1$. We add to this mean flow 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 $S = p/\rho^\gamma$

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

where $(\bar{x}, \bar{y}) = (x - x_0, y - y_0)$, $r^2 = \bar{x}^2 + \bar{y}^2$ and the vortex strength is $\epsilon = 5$. We regard the exact solution of this problem $W(x,y,t)$ as the passive convection of the vortex with the mean velocity and take the boundary conditions from $W(x,y,t)$ whenever needed. The number of boundary conditions is determined by the signs of the four eigenvalues λ_m which vary both in space and in time.

We use a second order Taylor expansion. In this 2D problem, least squares extrapolation is used instead of Lagrange extrapolation due to the stability issues mentioned in Section 2.6. We take the domain as a square $\Omega = (-0.5,1) \times (-0.5,1)$ or a disk $\Omega = \{(x, y): x^2 + y^2 < 0.5\}$. The density errors are listed in Tables 3.8 and 3.9. The designed third order accuracy is

Table 3.7
Errors of the 2D Burgers equation (3.16). $\Delta x = 2/N_x$, $\Delta y = 2/N_y$, $t = 0.15$.

$N_x = N_y$	On a square				On a disk			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
40	1.55E−04		9.86E−03		1.10E−04		1.77E−03	
80	1.06E−05	3.87	1.80E−03	2.46	7.24E−06	3.93	4.06E−04	2.12
160	4.93E−07	4.43	2.38E−04	2.91	4.65E−07	3.96	4.77E−05	3.09
320	3.47E−08	3.83	2.83E−05	3.08	3.63E−08	3.68	6.04E−06	2.98
640	2.72E−09	3.67	2.85E−06	3.31	4.10E−09	3.15	9.45E−07	2.68

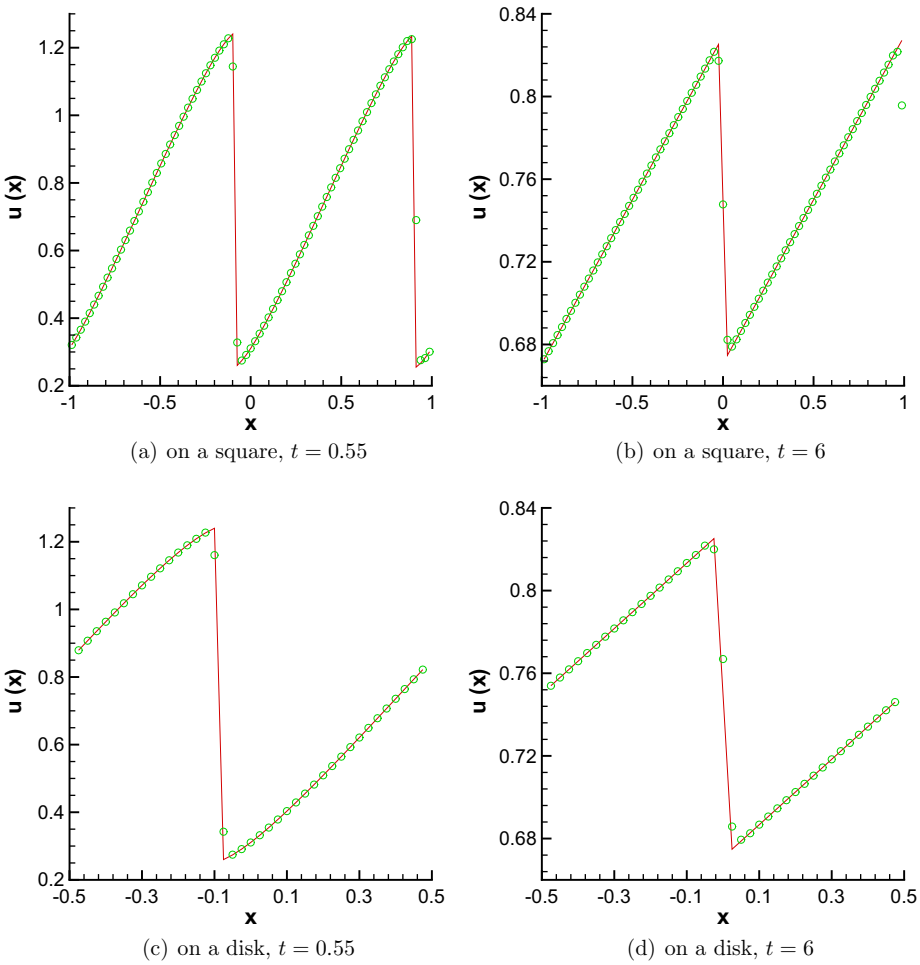


Fig. 3.6. 2D Burgers equation (3.16). $\Delta x = \Delta y = 1/40$. Cut along the diagonal. Solid line: exact solution; symbols: numerical solution.

achieved for this fully nonlinear problem. Both tables also contain additional computational cost due to our boundary treatment. We can see that our high order boundary treatment could be expensive, especially on a coarse mesh. In fact, for each ghost point, we have to solve three 4×4 linear systems or 5×4 linear least squares problems with right-hand side vectors requiring several numerical differentiations, which could be expensive showed by the 2D wave example.

Example 8. We are most interested in applying our method to the solid wall boundary conditions $(u,v) \cdot \boldsymbol{n} = 0$, when the wall is not aligned with the grid and can be curved. Our first example of this kind is the double Mach reflection problem [19]. This problem is initialized by sending a horizontally moving shock into a wedge inclined by a 30° angle. In order to impose the solid wall condition by the reflection technique, people usually solve an equivalent problem that puts the solid wall horizontal and puts the shock 60° angle inclined to the wall, see for example [8] and [15]. Another way to avoid the trouble of imposing boundary conditions is to use a multidomain WENO method [14].

With the use of our method, we are able to solve the original problem with a uniform mesh in a single domain. The computational domain is shown in Fig. 3.7(a), together with some of the grid points near the wall which indicate that the wall is not aligned with the grid. Initially a right-moving Mach 10 shock is positioned at (0,0) making an angle of 90° with the

Table 3.8
Density errors and cost due to boundary treatment (in terms of ratio between the CPU time of boundary treatment and the total CPU time) of the vortex evolution problem on a square. The vortex is initially positioned at (0,0). $\Delta x = 1.5/N$ and $t = 1$.

$N_x \times N_y$	L^1 error	Order	L^∞ error	Order	Cost
40×40	3.41E−06		1.81E−05		0.56
80×80	2.38E−07	3.84	1.86E−06	3.28	0.40
160×160	2.08E−08	3.52	1.83E−07	3.34	0.25
320×320	2.40E−09	3.11	1.97E−08	3.22	0.14

Table 3.9
Density errors and cost due to boundary treatment (in terms of ratio between the CPU time of boundary treatment and the total CPU time) of the vortex evolution problem on a disk. The vortex is initially positioned at (0.3, 0.3). $\Delta x = 2/N$ and $t = 0.1$.

$N_x \times N_y$	L^1 error	Order	L^∞ error	Order	Cost
80×80	1.21E−07		2.29E−05		0.56
160×160	4.91E−09	4.62	1.29E−06	4.15	0.54
320×320	4.01E−10	3.61	1.27E−07	3.34	0.45
640×640	4.32E−11	3.21	1.60E−08	2.99	0.32

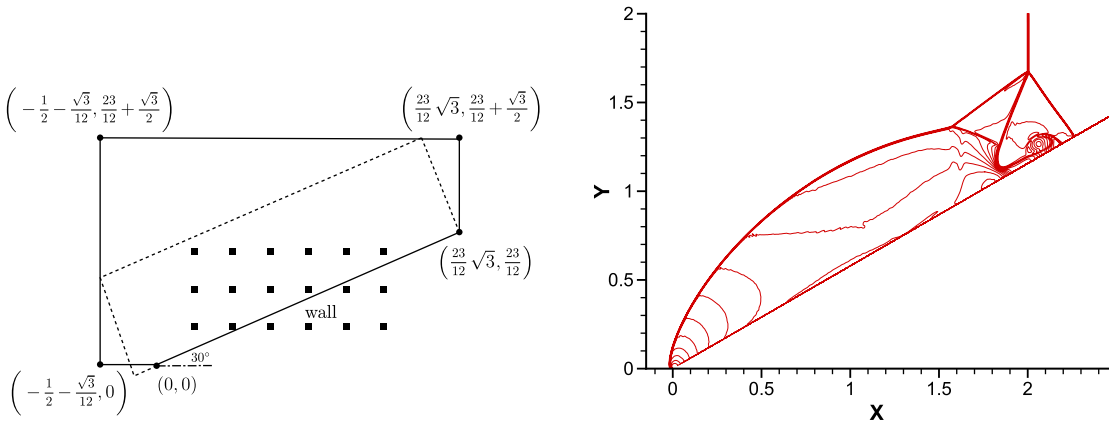


Fig. 3.7. Left: The computational domain (solid line) of the double Mach reflection problem. The dashed line indicates the computational domain used in [8] and [15]. The square points indicate some of the grid points near the wall. Illustrative graph, not to scale. Right: Density contour of double Mach reflection, 30 contours from 1.731 to 20.92. $\Delta x = \Delta y = \frac{1}{320}$.

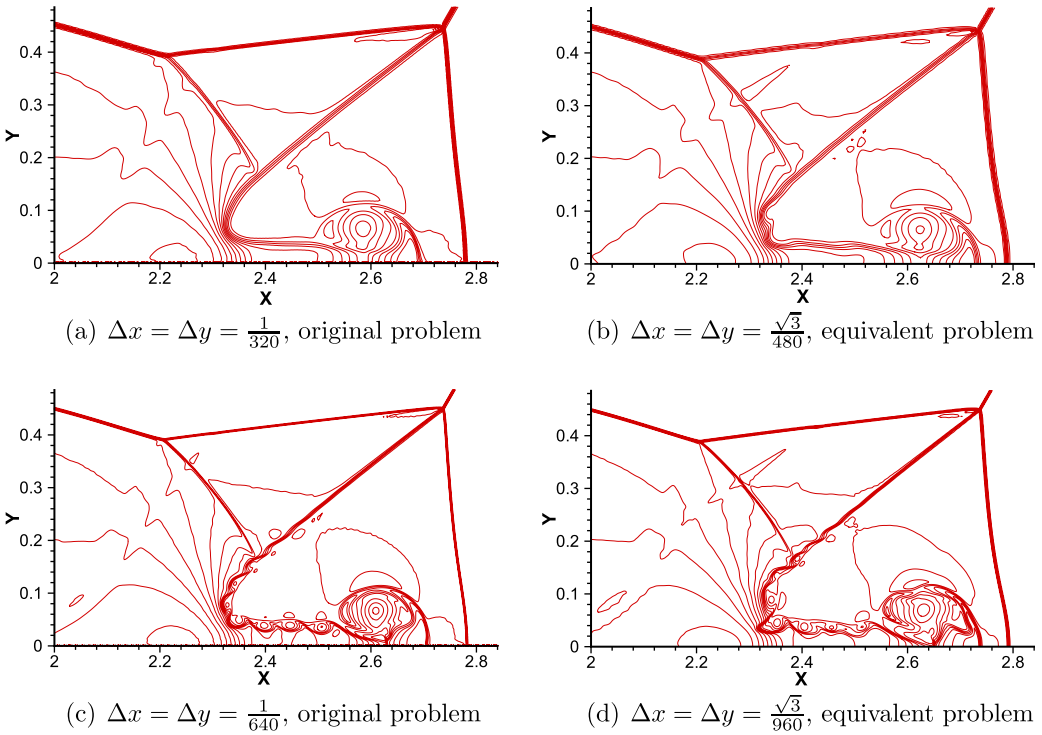


Fig. 3.8. Density contours of double Mach reflection, 30 contours from 1.731 to 20.92. Zoomed-in near the double Mach stem. The plots in the left column are rotated and translated for comparison.

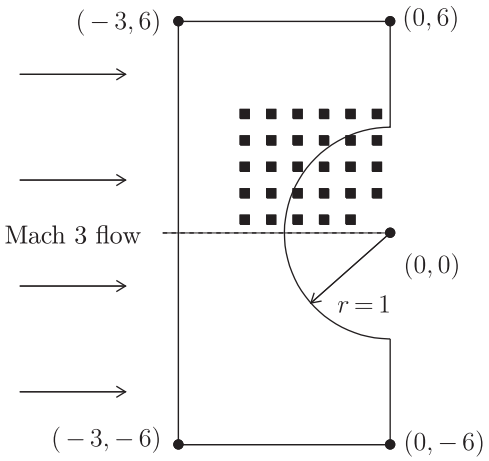
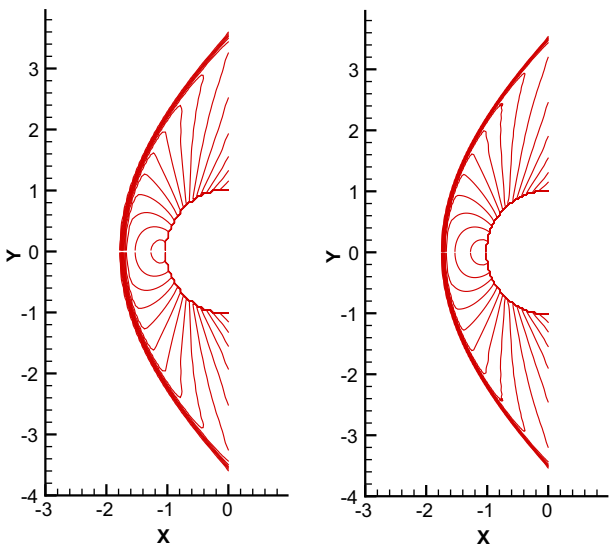


Fig. 3.9. Physical domain of flow past a cylinder. The square points indicate some of the grid points near the cylinder. Illustrative sketch, not to scale.

x-axis. At $y = 0$, the exact postshock condition is imposed. At the top boundary $y = \frac{23}{12} + \frac{\sqrt{3}}{2}$, the flow values are set to describe the exact motion of the Mach 10 shock. At the wall, we use a second order Taylor expansion with the inverse Lax-Wendroff procedure and the WENO type extrapolation. Fig. 3.7(b) shows the density contour on a mesh with $\Delta x = \Delta y = \frac{1}{320}$. A zoomed-in region near the double Mach stem is shown in Fig. 3.8(a). We rotate and translate the region for ease of comparison. In Fig. 3.8(b), we show the result of the regular fifth order WENO scheme for the equivalent problem with a regular reflective boundary condition on a mesh with a comparable size. Figs. 3.8(c) and 3.8(d) show the density contours on a refined mesh. We can see that the results of our boundary treatment are very similar to those obtained by the reflection technique. The slight difference comes perhaps from the fact we impose the no-penetration condition strongly while the reflection technique imposes it weakly.

Example 9. Our final example involves a curved wall which is a circular cylinder of unit radius positioned at the origin on a x–y plane. The problem is initialized by a Mach 3 flow moving toward the cylinder from the left. In order to impose the solid wall boundary condition at the surface of the cylinder by the reflection technique, a particular mapping from the unit square to the physical domain is used in [8]. Using our method, we are able to solve this problem directly in the physical domain, which is shown in Fig. 3.9, together with some of the grid points near the cylinder which indicate the wall cuts the grid in an arbitrary fashion. Our computational domain is the upper half of the physical domain, due to the symmetry of this problem. At $y = 0$, the reflection technique is used. At the left inflow boundary $x = -3$, the uniform far-field data is imposed. At the top boundary $y = 6$ and the right boundary $x = 0$, constant extrapolation is used because of the hyperbolic nature of this problem.



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

Fig. 3.10. Pressure contour of flow past a cylinder, 20 contours from 2 to 15.

A second order Taylor expansion with the inverse Lax-Wendroff procedure and the WENO type extrapolation is used at the surface of the cylinder. Notice that we impose the no-penetration condition strongly. Thus we should modify the initial condition so that it is compatible with the solid wall condition. The pressure contour is shown in Fig. 3.10(a) with $\Delta x = \Delta y = \frac{1}{20}$ and in Fig. 3.10(b) with $\Delta x = \Delta y = \frac{1}{40}$. We can see the bow shock is well-captured by our method.

4. Concluding remarks

In this paper, we develop a high order numerical boundary condition for solving hyperbolic conservation laws with finite difference methods. It is based on a Cartesian mesh, which is very challenging for boundary treatment because of the wide stencil of the interior scheme and the fact that the physical boundary is not necessarily aligned with the mesh. Our method consists of the inverse Lax-Wendroff type procedure for inflow boundary conditions and extrapolation for outflow boundary conditions. The idea of the inverse Lax-Wendroff type procedure comes from the original Lax-Wendroff scheme. We repeatedly utilize the PDE to write the normal derivatives of $\boldsymbol{U}$ in terms of the time derivatives and tangential derivatives of $\boldsymbol{U}$, both of which are known on the inflow boundary. At the outflow boundary, we use Lagrange extrapolation or least squares extrapolation if the solution is smooth, or WENO type extrapolation if a shock is close to the boundary. A variety of numerical examples illustrate that our method is high order accurate and is capable of treating shocks going through the boundaries. Moreover, our method performs well when applied to the solid wall boundary conditions. The additional computational cost of our boundary treatment is negligible for one-dimensional problems. However, it could be a significant portion of the total computational cost for solving two-dimensional Euler equations due to the extensive numerical differentiation involved.

We have shown the linear stability of the semi-discrete scheme for one-dimensional scalar problems. Our method seems to be stable for one-dimensional examples in our numerical experiments. In two-dimensional examples with smooth solutions, a mild instability is discovered at subsonic outflow boundaries. This instability results from high order Lagrange extrapolation. Least squares extrapolation is used instead to stabilize the scheme in our numerical tests. To completely understand the mechanism of the instability caused by extrapolation, a GKS analysis for 2D problems should be done in the future. The GKS theory can also help us construct a stable scheme. The methodology can be applied in three dimensions straightforwardly. However, the algebra will be more heavy, and it remains to be seen what the ratio of the computational cost for such boundary treatment over that for the inner scheme would be.

References

[1] M.J. Berger, C. Helzel, R.J. Leveque, h-Box methods for the approximation of hyperbolic conservation laws on irregular grids, SIAM Journal on Numerical Analysis 41 (2003) 893–918.
[2] 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 Journal on Scientific Computing 16 (1995) 1241–1252.
[3] M. Goldberg, On a boundary extrapolation theorem by Kreiss, Mathematics of Computation 31 (1977) 469–477.
[4] S. Gottlieb, D.I. Ketcheson, C.-W. Shu, High order strong stability preserving time discretizations, Journal of Scientific Computing 38 (2009) 251–289.
[5] B. Gustafsson, H.-O. Kreiss, A. Sundström, Stability theory of difference approximations for mixed initial boundary value problems. II. Mathematics of Computation 26 (1972) 649–686.
[6] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, Journal of Computational Physics 71 (1987) 231–303.
[7] L. Huang, C.-W. Shu, M. Zhang, Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation, Journal of Computational Mathematics 26 (2008) 336–346.
[8] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.
[9] H.-O. Kreiss, N.A. Petersson, A second order accurate embedded boundary method for the wave equation with Dirichlet data, SIAM Journal on Scientific Computing 27 (2006) 1141–1167.
[10] H.-O. Kreiss, N.A. Petersson, J. Yström, Difference approximations for the second order wave equation, SIAM Journal on Numerical Analysis 40 (2002) 1940–1967.
[11] H.-O. Kreiss, N.A. Petersson, J. Yström, Difference approximations of the Neumann problem for the second order wave equation, SIAM Journal on Numerical Analysis 42 (2004) 1292–1323.
[12] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, Journal of Computational Physics 211 (2006) 492–512.
[13] P.D. Lax, B. Wendroff, Systems of conservation laws, Communications in Pure and Applied Mathematics 13 (1960) 217–237.
[14] K. Sebastian, C.-W. Shu, Multidomain WENO finite difference method with interpolation at subdomain interfaces, Journal of Scientific Computing 19 (2003) 405–438.
[15] J. Shi, Y.-T. Zhang, C.-W. Shu, Resolution of high order WENO schemes for complicated flow structures, Journal of Computational Physics 186 (2003) 690–696.
[16] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.
[17] B. Sjögreen, N.A. Petersson, A Cartesian embedded boundary method for hyperbolic conservation laws, Communications in Computational Physics 2 (2007) 1199–1219.
[18] J.C. Strikwerda, Initial boundary value problems for the method of lines, Journal of Computational Physics 34 (1980) 94–107.
[19] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.
[20] T. Xiong, M. Zhang, Y.-T. Zhang, C.-W. Shu, Fast sweeping fifth order WENO scheme for static Hamilton–Jacobi equations with accurate boundary treatment, Journal of Scientific Computing (2010), doi:10.1007/s10915-010-9345-6.