Inverse Lax-Wendroff procedure for numerical boundary treatment of hyperbolic equations

Chi-Wang Shu\textsuperscript{1} and Sirui Tan\textsuperscript{2}

Abstract

We discuss a high order accurate numerical boundary condition for solving hyperbolic conservation laws on fixed Cartesian grids, while the physical domain can be arbitrarily shaped and moving. Compared with body-fitted meshes, the biggest advantage of Cartesian grids is that the grid generation is trivial. The challenge is however that the physical boundary does not usually coincide with grid lines. The wide stencil of the high order interior scheme makes a stable boundary treatment even harder to realize. There are two main ingredients of our method. The first one is an inverse Lax-Wendroff procedure for inflow boundary conditions and the other one is a robust and high order accurate extrapolation for outflow boundary conditions. Our method is high order accurate, stable, and easy to implement. It has been successfully applied to simulate interactions between compressible inviscid flows and rigid (static or moving) bodies with complex geometry.

\textbf{Keywords:} Cartesian mesh; compressible inviscid flow; extrapolation; hyperbolic conservation law; inverse Lax-Wendroff procedure; no-penetration condition; numerical boundary condition

\textsuperscript{1}Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: shu@dam.brown.edu.

\textsuperscript{2}Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: sirui.tan@alumni.brown.edu.
1 Introduction

We consider high order finite difference methods for solving hyperbolic equations involving complex static for moving geometry. For such problems, body-fitted meshes which conform to the geometry are often used because of the ease of imposing boundary conditions. In the context of finite difference methods, body-fitted grids are usually generated by a curvilinear transformation that maps the physical domain with complex geometry to a rectangular computational domain which can be easily gridded. The design of a curvilinear transformation could be difficult for geometrically complicated domains. In addition, the partial differential equations in the transformed domain are usually more complicated than those in the physical domain, leading to increased computational costs. Alternatively, Cartesian grids can be used to discretize the physical domain regardless of its geometry. The biggest challenge is however to accurately impose the boundary conditions while retaining the stability. The challenge mainly results from two facts. First, the physical boundary does not necessarily coincide with grid lines and can intersect the Cartesian grids in an arbitrary fashion. Secondly, high order interior schemes need proper treatment of several ghost points near the boundary because of the wide numerical stencils.

There are many successful numerical methods based on Cartesian grids. The immersed boundary method (IBM) introduced by Peskin (1972) is widely used to solve incompressible flows in complicated time-varying domains. The IBM was extended for compressible viscous flows (Palma et al., 2006; de Tullio et al., 2007; Ghias et al., 2007). An immersed interface method was developed for elliptic equations (LeVeque and Li, 1994, 1997) and for streamfunction-vorticity equations (LeVeque and Calhoun, 2001). We emphasize that high order numerical boundary conditions for hyperbolic equations are more difficult than elliptic or convection-diffusion type equations mentioned above in the sense that there may be strong discontinuities passing through the boundaries in the hyperbolic case. The boundary treatment should be robust when strong shock waves reflect off rigid walls.

The most popular way to impose boundary conditions at rigid boundaries for compressible
flows is the reflection technique, where extra rows of ghost points are added. All interior solution components are reflected symmetrically to ghost states except for the normal velocity which is negated. This method works well when the rigid boundary is straight and positioned at half grid points. It might lead to large errors otherwise. The inaccuracy of the approach when applied to curved geometry was demonstrated by Moretti (1969).

For finite volume methods, boundary lines may cut Cartesian cells into irregular ones with volumes orders of magnitude smaller than the regular grid cells, leading to a severe time step restriction (the so-called “cut-cell problem”). The $h$-box method (Berger et al., 2003; Helzel et al., 2005) was developed to overcome this problem. The basic idea is to approximate numerical fluxes at the interface of a small cell based on initial values specified over regions of a regular length $h$. For one-dimensional (1D) convection equations, the $h$-box method is shown to be conservative and stable on arbitrary irregular grids (Berger et al., 2003). Numerical results confirm the same conclusion for Euler equations (Helzel et al., 2005). Another method to avoid the small-cell problem for Euler equations is to maintain cut boundary cells as whole (ghost) cells and obtain ghost cell values by reflecting locally the flow field around the boundary which is approximated by straight lines (Forrer and Jeltsch, 1998). This method is stable and applicable to any finite volume method, but conservation at the boundary is violated although this error may be relatively small. For Euler equations involving moving geometry, an idea similar to that of Forrer and Jeltsch (1998) was developed (Forrer and Berger, 1999) but the ghost cell values are obtained by a mirror flow extrapolation. A cell merging technique combined with dimension splitting was developed by Falcovitz et al. (1997). Shyue (2008) proposed a moving-boundary tracking algorithm based on finite volume wave-propagation method. The method is stable even if there are very small cut cells near the tracked interface. Arienti et al. (2003) developed an Eulerian-Lagrangian coupling scheme. Hu et al. (2006) developed a conservative interface method based on the level set technique. In terms of accuracy, all the finite volume schemes mentioned above are at most second order. In particular, the errors at the boundaries
sometimes fall short of second order (Arienti et al., 2003; Forrer and Jeltsch, 1998; Hu et al., 2006).

For finite difference methods, a second order accurate Cartesian embedded boundary method was developed to solve the wave equations with Dirichlet or Neumann boundary conditions (Kreiss and Petersson, 2006; Kreiss et al., 2002, 2004) and to solve hyperbolic conservation laws (Sjögreen and Petersson, 2007). The basic idea is to assign values to ghost points outside the domain through extrapolation. To avoid oscillations near shock waves, slope limiters are combined with extrapolation (Sjögreen and Petersson, 2007). This approach was extended from second order to fifth order by using Lagrange extrapolation with a filter for the detection of discontinuities (Baeza et al., 2016a) and by a weighted extrapolation technique (Baeza et al., 2016b). Appelö and Petersson (2012) modified the embedded boundary method by assigning values to boundary points inside the domain via interpolation. This method is fourth order accurate and is based on a compact finite difference scheme. Fisher et al. (2011) developed stable boundary conditions in $L^2$ norm for fourth-order energy stable weighted essentially non-oscillatory (WENO) schemes (Yamaleev and Carpenter, 2009). The boundary conditions maintain, whenever possible, the WENO stencil biasing properties and satisfy the summation-by-part operator convention. A distinct feature of the schemes is that they require nonuniform flux interpolation points near the boundaries.

A Lax-Wendroff type boundary condition procedure was introduced by Huang et al. (2008) for solving static Hamilton-Jacobi equations with a third order finite difference method. The boundary condition was extended to fifth order (Xiong et al., 2010) and to discontinuous Galerkin method (Zhang et al., 2011) for the same type of problems. The idea is to repeatedly use the partial differential equations (PDEs) 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 article, we discuss systematically the
extension of this procedure to solve time dependent hyperbolic equations involving complex static or moving geometry. For time dependent problems, the boundary treatment procedure is in essence repeatedly using the PDEs to convert normal spatial derivatives to time and tangential derivatives of the given boundary condition. It is in some sense an inverse to the standard Lax-Wendroff procedure (Lax and Wendroff, 1960), in which the PDEs are repeatedly used to convert time derivatives to spatial derivatives when discretizing the PDEs in time with high order accuracy. We therefore refer to our method as the inverse Lax-Wendroff (ILW) procedure. This procedure was first proposed by Goldberg and Tadmor (1978, 1981) for analyzing numerical boundary conditions of linear hyperbolic equations in one dimension with boundaries aligned with grid lines. Lombard et al. (2008) applied a similar idea to arbitrary-shaped free boundaries in finite difference schemes for linear elastic waves. We extend the methods of Goldberg and Tadmor (1978, 1981) and Lombard et al. (2008) to nonlinear hyperbolic problems. In particular, strong discontinuities near the boundaries, which are absent in linear elastic waves, are handled by a high order WENO extrapolation to prevent overshoot or undershoot. Moreover, we propose a simplified and improved implementation, which uses the relatively complicated ILW procedure only for the evaluation of the first order normal derivatives. High order WENO extrapolation is used for all other derivatives, regardless of the direction of the local characteristics and the smoothness of the solution. This makes the implementation of a very high order boundary treatment practical for two-dimensional (2D) nonlinear systems with source terms. For no-penetration boundary condition of compressible inviscid flows, a further simplification is discussed, in which the evaluation of the tangential derivatives involved in the ILW procedure is avoided.

Compared with the Cartesian embedded boundary method (e.g. Sjögreen and Petersson, 2007), the disadvantage of the ILW procedure is that the formulation and implementation depend on the PDEs being solved. On the other hand, the linear stability of the ILW procedure with up to thirteenth order of accuracy has been guaranteed for 1D problems through rigorous analysis (Li et al., 2016). To the best of our knowledge, the linear stability
of the Cartesian embedded boundary method is unknown for schemes with higher than second order of accuracy.

In this article, we present the ILW procedure for numerical boundary conditions in the context of finite difference WENO schemes (Jiang and Shu, 1996) for solving hyperbolic conservation laws. The methodology has been applied to other types of problems or schemes, such as compact finite difference schemes (Vilar and Shu, 2015) and Boltzmann equations (Filbet and Yang, 2013).

2 Problem description and interior schemes

We consider hyperbolic conservation laws possibly with source terms for $U = U(x, y, t) \in \mathbb{R}^2$

\[
\begin{aligned}
&U_t + F(U)_x + G(U)_y = S(U) \quad (x, y) \in \Omega(t), \quad t > 0, \\
&U(x, y, 0) = U_0(x, y) \quad (x, y) \in \Omega(t),
\end{aligned}
\]

on a bounded domain $\Omega(t)$ with appropriate boundary conditions prescribed on $\Gamma(t) = \partial \Omega(t)$ at time $t$. We assume there is an analytical expression for the geometry of $\Gamma(t)$. $\Omega(t)$ is covered by a uniform Cartesian mesh with mesh size $\Delta x = \Delta y = h$, but the boundary $\Gamma(t)$ does not need to coincide with any grid lines. The semi-discrete approximation of (1) is given by

\[
\frac{d}{dt} U_{i,j}(t) = -\frac{1}{h} \left( \hat{F}_{i+1/2,j} - \hat{F}_{i-1/2,j} \right) \\
- \frac{1}{h} \left( \hat{G}_{i,j+1/2} - \hat{G}_{i,j-1/2} \right) + S(U_{i,j}(t)),
\]

where $\hat{F}_{i+1/2,j}$ and $\hat{G}_{i,j+1/2}$ are the numerical fluxes. We use the third order total variation diminishing Runge-Kutta method (Shu and Osher, 1988) to integrate the system of ordinary differential equations (2) in time.

Our boundary treatment method is applicable to general finite difference schemes. To be more precise for this article, we use the fifth order finite difference WENO scheme with the Lax-Friedrichs flux splitting to form the numerical fluxes $\hat{F}_{i+1/2,j}$ and $\hat{G}_{i,j+1/2}$ in (2) (Jiang and Shu, 1996). The scheme requires a seven point stencil in both $x$ and $y$ directions. Near $\Gamma(t)$ where the numerical stencil is partially outside of $\Omega(t)$, up to three ghost points are
needed in each direction. In some cases of moving boundaries, up to four ghost points may be needed in each direction. We concentrate on how to define the values of the ghost points at time level $t = t_n$ in Section 3 and Section 4 for static and moving geometry, respectively.

3 Numerical boundary conditions for static geometry

3.1 One-dimensional scalar conservation laws: smooth solutions

To illustrate the essential idea of the ILW procedure, we use 1D scalar conservation laws as an example

$$
\begin{align*}
&u_t + f(u)_x = 0 & x \in (a, b), & t > 0, \\
&u(a, t) = g(t) & t > 0, \\
&u(x, 0) = u_0(x) & x \in [a, b].
\end{align*}
$$

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

We discretize the interval $(a, b)$ by a uniform mesh

$$
a + C_a h = x_0 < x_1 < \ldots < x_N = b - C_b h,
$$

where $0 \leq C_a < 1$ and $0 \leq C_b < 1$. Notice that both $x_0$ and $x_N$ are not necessarily located on the boundary, which is chosen this way on purpose since it is usually impossible to align boundary with grid points in a 2D domain with complex geometry. We assume $u_0 \approx u(x_0, t_n), \ldots, u_N \approx u(x_N, t_n)$ have been updated from time level $t_{n-1}$ to time level $t_n$.

Here we suppress the $t_n$ dependence in $u_0, \ldots, u_N$ without causing any confusion.

3.1.1 Inverse Lax-Wendroff procedure for inflow boundary conditions

At the inflow boundary $x = a$, we construct a $s$th order approximation of ghost point values $u_{-3}, \ldots, u_{-1}$ by a Taylor expansion

$$
u_j = \sum_{k=0}^{s-1} \frac{(x_j - a)^k}{k!} u_{L}^{(k)}, \quad j = -3, \ldots, -1,
$$

(3)
where \( u_{L}^{(k)} \) is a \((s-k)\)th order approximation of \( \left. \frac{\partial^{k}u}{\partial x^{k}} \right|_{x=a, t=t_{n}} \). We impose \( u_{L}^{(0)} = g(t_{n}) \). To obtain the approximation of the spatial derivative \( \left. \frac{\partial u}{\partial x} \right|_{x=a, t=t_{n}} \), we utilize the PDE

\[
 u_{t} + f'(u)u_{x} = 0
\]

and evaluate it at \( x = a, t = t_{n} \). We impose

\[
 u_{L}^{(1)} = u_{LW}^{(1)} = -\frac{u_{t}(a, t_{n})}{f'(u(a, t_{n}))} = -\frac{g'(t_{n})}{f'(g(t_{n}))},
\]

where \( f'(g(t_{n})) \) is bounded away from zero by the assumption that \( x = a \) is an inflow boundary. Differentiating the PDE with respect to time yields

\[
 u_{tt} + f''(u)u_{t}u_{x} + f'(u)u_{xt} = 0. \tag{4}
\]

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 (4), we obtain an equation for \( u_{xx} \)

\[
 u_{tt} + f''(u)u_{t}u_{x} - f'(u)f''(u)u_{x}^{2} - f'(u)^{2}u_{xx} = 0. \tag{5}
\]

Solving (5) for \( u_{xx} \) and evaluating it at \( x = a, t = t_{n} \), we impose

\[
 u_{L}^{(2)} = u_{LW}^{(2)} = \frac{f'(g(t_{n}))g''(t_{n}) - 2f''(g(t_{n}))g'(t_{n})^{2}}{f'(g(t_{n}))^{3}}.
\]

Following the same procedure, we can impose values of \( u_{L}^{(k)} = u_{LW}^{(k)}, \ k = 1, \ldots, s - 1. \)

The idea of converting time derivatives to spatial derivatives by repeatedly using the PDE comes from the original Lax-Wendroff scheme (Lax and Wendroff, 1960). Since we convert spatial derivatives to time derivatives, our method is called the \textit{inverse} Lax-Wendroff procedure. We remark that this procedure is independent of 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 discontinuities going through the boundary, \( g(t) \) is discontinuous. The stencil used for numerical differentiation should not contain any discontinuity. For example, an essentially non-oscillatory (ENO) procedure (Harten et al., 1997) or a WENO procedure (Jiang and Shu, 1996) can be used for this numerical differentiation.

### 3.1.2 Lagrange extrapolation for outflow boundary conditions

At the outflow boundary \( x = b \), values of ghost points \( u_{N+1}, \ldots, u_{N+3} \) are also approximated by a \((s - 1)\)th order Taylor expansion

\[
    u_j = \sum_{k=0}^{s-1} \frac{(x_j - b)^k}{k!} u_R^{s(k)}, \quad j = N + 1, \ldots, N + 3,
\]

where \( u_R^{s(k)} \) is a \((s - k)\)th order approximation of \( \frac{\partial^k u}{\partial x^k} \bigg|_{x=b, t=t_n} \). At the outflow boundary, \( u_R^{s(k)} \) should be imposed by the values of the interior points, \( u_0, \ldots, u_N \), because of the outgoing characteristics, even if a boundary condition is improperly prescribed. If \( u(x) \) is smooth near the boundary, \( u_R^{s(k)} \) can be easily obtained by

\[
    u_R^{s(k)} = \frac{d^k p_{R}^{s-1}(x)}{dx^k} \bigg|_{x=b},
\]

where \( p_{R}^{s-1}(x) \) is a Lagrange polynomial of degree at most \( s - 1 \) interpolating \( u_j, j = N - s + 1, \ldots, N \). An explicit formula for the values of ghost points is

\[
    \sum_{k=0}^{s} \frac{s!}{k!(s-k)!} (-1)^k u_{j-k} = 0, \quad j = N + 1, \ldots, N + 3. \tag{6}
\]

For example, for \( s = 5 \)

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

### 3.1.3 Simplified inverse Lax-Wendroff procedure

We recall that the spatial derivatives \( u_L^{s(k)}, k \geq 1 \), in the Taylor expansion (3) are obtained by repeated use of the PDE at the inflow boundary. Obviously, the algebra of converting derivatives of order higher than or equal to two can be very heavy if the PDE is complicated, which is usually the case if we consider 2D fully nonlinear systems (1). The complicated
algebra would prevent us from implementing higher than third order accurate boundary
treatment for 2D Euler equations (Tan and Shu, 2010, 2011), although our method is designed
to achieve arbitrarily high order accuracy for general equations with source terms.

Alternatively, we may obtain the spatial derivatives $u^{*(k)}_{L}$ using extrapolation given by

$$u^{*(k)}_{L} = u^{(k)}_{EXT} = \frac{d^{k}p^{s-1}_{L}(x)}{dx^{k}} \bigg|_{x=a},$$

where $p^{s-1}_{L}(x)$ is a Lagrange polynomial of degree at most $s-1$ interpolating $u_{j}$, $j = 0, \ldots, s-1$. A natural way to combine the ILW procedure and extrapolation is that

$$u^{*(k)}_{L} = \begin{cases} u^{(k)}_{ILW} & k \leq k_{s} - 1, \\ u^{(k)}_{EXT} & k \geq k_{s}. \end{cases}$$

Namely, we use the ILW procedure to obtain the spatial derivatives of order up to $k_{s} - 1$; we
switch to simple extrapolation when the algebra of the ILW procedure becomes prohibitive
for spatial derivatives of order greater than or equal to $k_{s}$. This simplified version of the
ILW procedure is referred to as the simplified inverse Lax-Wendroff procedure (Tan et al.,
2012; Tan and Shu, 2013). A similar idea is proposed by Qiu and Shu (2003) for the Lax-
Wendroff time discretization, in which they perform a WENO approximation only for the
reconstruction of the fluxes to the first order time derivative to avoid spurious oscillations
and use the inexpensive central difference approximation for the reconstruction of the higher
order time derivatives.

3.1.4 Linear stability

Li et al. (2016) performed the linear stability analysis of the ILW and the simplified ILW
boundary treatments for 1D initial-boundary value problems. The interior finite difference
schemes used in the analysis are WENO schemes with linear weights, which are essentially
high-order upwind-biased schemes. For the simplified ILW procedure, the stability depends
on $k_{s}$, which is the number of Taylor expansion terms obtained by the ILW procedure.
Simplified ILW maintains stability with the same Courant-Friedrichs-Lewy (CFL) number
as the periodic boundary case for any boundary locations when $k_{s}$ is at least taken as $(k_{s})_{\text{min}}.$
When $k_s$ is less than $(k_s)_{\text{min}}$, simplified ILW may still be stable but with more restrictive CFL conditions. For example, for the fifth order simplified ILW procedure, $(k_s)_{\text{min}} = 3$, which means we have to utilize the ILW procedure for up to second order spatial derivatives in order to use the maximum allowed CFL number of the periodic boundary case. If we utilize the ILW procedure only for the first order spatial derivative, i.e., $k_s = 2$, we still have a stable boundary treatment. However, the maximum allowed CFL number becomes 1.02, which is compared with the maximum allowed CFL number 1.43 for the periodic boundary case.

3.2 One-dimensional scalar conservation laws: solutions containing discontinuities

When a shock is very close to the outflow boundary, high order Lagrange extrapolation may lead to severe overshoot or undershoot near the discontinuity. See Figure 1(a) for an example. In this situation, we prefer a possibly lower order accurate but more robust extrapolation. The fifth order WENO extrapolation is developed for this purpose.

We need to obtain $u^*(k)_R$, a $(5 - k)$th order approximation of the $k$th order derivative of $u(x)$ at the boundary, using the grid point values in the interior of the domain $u_j, j = N - 4, \ldots, N$. We consider the stencil $S_1 = \{x_{N-4}, \ldots, x_N\}$ as five candidate substencils $S_r = \{x_{N-r}, \ldots, x_N\}, r = 0, \ldots, 4$. On each stencil, we can easily construct a Lagrange polynomial $p_r(x)$ of degree at most $r$ such that $p_r(x_j) = u_j, j = N - r, \ldots, N$. Suppose $u$ is smooth on $S_1$, $u^*(k)_R$ can be extrapolated by

$$u^*(k)_R = \sum_{r=0}^{4} \gamma_r \frac{d^k p_r(x)}{dx^k} \bigg|_{x=b} ,$$

where $\gamma_0 = h^4$, $\gamma_1 = h^3$, $\gamma_2 = h^2$, $\gamma_3 = h$, and $\gamma_4 = 1 - \sum_{r=0}^{3} \gamma_r$.

Borrowing the idea of WENO reconstruction (Jiang and Shu, 1996), we look for WENO type extrapolation in the form

$$u^*(k)_R = \sum_{r=0}^{4} \omega_r \frac{d^k p_r(x)}{dx^k} \bigg|_{x=b} ,$$ (7)
Figure 1: Burgers equation \( u_t + \left( \frac{1}{2} u^2 \right)_x = 0 \) with initial condition \( u(x,0) = 0.25 + 0.5 \sin(\pi x), x \in [-1,1] \). \( h = 1/40 \) and \( t = 7.8 \). The outflow boundary condition is imposed by (a) Lagrange extrapolation and (b) WENO extrapolation. Solid line: exact solution; Symbols: numerical solution.

where \( \omega_r \) are the nonlinear weights depending on the value of \( u_j \). In the case that \( u \) is smooth in \( S_4 \), we would like to have

\[
\omega_0 = O(h^4), \hspace{1em} \omega_1 = O(h^3), \hspace{1em} \omega_2 = O(h^2), \hspace{1em} \omega_3 = O(h) \text{ and } \omega_4 = 1 - \sum_{r=0}^{3} \omega_r
\]  

such that (7) is \((5 - k)\)th order accurate. The nonlinear weights \( \omega_r \) are defined by

\[
\omega_r = \frac{\tilde{\omega}_r}{\sum_{\nu=0}^{4} \tilde{\omega}_\nu}
\]

with

\[
\tilde{\omega}_r = \frac{\gamma_r}{(\varepsilon + \beta_r)^q},
\]

where \( \varepsilon = 10^{-6}, q \geq 3 \) is an integer, and \( \beta_r \) are the smoothness indicators determined by

\[
\beta_0 = h^2,
\]

\[
\beta_r = \sum_{l=1}^{r} \int_{b}^{b+h} h^{2l-1} \left( \frac{d^l}{dx^l} P_r(x) \right)^2 dx, \hspace{1em} r = 1, \ldots, 4.
\]
The explicit expressions for the smoothness indicators $\beta_1$ and $\beta_2$ are

$$\beta_1 = (u_{N-1} - u_N)^2,$$

$$\beta_2 = (61u_N^2 + 160u_{N-1}^2 + 74u_Nu_{N-2} + 25u_{N-2}^2 - 196u_{N-1}u_N + 124u_{N-1}u_{N-2})/12.$$  

The expressions for $\beta_3$ and $\beta_4$ are tedious but can be easily derived by symbolic computation softwares.

Note that a drawback of the smoothness indicators defined in (10) is that they are scale dependent. A proper scaling factor proposed by Filbet and Yang (2013) is that

$$\beta_{r,scaled} = \frac{1}{\varepsilon + \sum_{l=0}^{r} u_N^{2l}} \beta_r, \quad r = 1, \ldots, 4.$$  

The undershoot shown in Figure 1(a) is well-controlled by this robust extrapolation, see Figure 1(b). Moreover, the fifth order accuracy of smooth solutions is maintained, see the errors of the initial-boundary value problem of the Burgers equation in Table 1. Therefore, we use the more robust WENO extrapolation from now on, regardless of the smoothness of the solution.

Table 1: Errors of the Burgers equation $u_t + \left(\frac{1}{2}u^2\right)_x = 0$ with initial condition $u(x,0) = 0.25 + 0.5 \sin(\pi x), x \in [-1,1]$. $\Delta t = O(h^{5/3})$ and $t = 0.3$. The inflow boundary condition is imposed by the ILW procedure; the outflow boundary condition is imposed by WENO extrapolation.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$L^1$ error</th>
<th>order</th>
<th>$L^\infty$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/40</td>
<td>3.10E-06</td>
<td></td>
<td>1.35E-05</td>
<td></td>
</tr>
<tr>
<td>1/80</td>
<td>1.31E-07</td>
<td>4.57</td>
<td>6.51E-07</td>
<td>4.38</td>
</tr>
<tr>
<td>1/160</td>
<td>3.97E-09</td>
<td>5.05</td>
<td>2.68E-08</td>
<td>4.60</td>
</tr>
<tr>
<td>1/320</td>
<td>1.02E-10</td>
<td>5.29</td>
<td>8.34E-10</td>
<td>5.00</td>
</tr>
<tr>
<td>1/640</td>
<td>2.86E-12</td>
<td>5.15</td>
<td>2.62E-11</td>
<td>5.00</td>
</tr>
<tr>
<td>1/1280</td>
<td>1.04E-13</td>
<td>4.78</td>
<td>9.09E-13</td>
<td>4.85</td>
</tr>
</tbody>
</table>

3.3 Two-dimensional Euler equations in static geometry

We consider 2D compressible Euler equations in static geometry

$$U_t + F(U)_x + G(U)_y = 0, \quad (x,y) \in \Omega, \quad t > 0,$$  

(11)
where

\[
U = \begin{pmatrix}
U_1 \\
U_2 \\
U_3 \\
U_4
\end{pmatrix} = \begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
E
\end{pmatrix}, \quad F(U) = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
u(E + p)
\end{pmatrix}, \quad G(U) = \begin{pmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
v(E + p)
\end{pmatrix},
\]

with appropriate boundary conditions and initial conditions. \( \rho, u, v, p, \) 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 the ratio of specific heats \( \gamma = 1.4 \) for air at ordinary temperatures.

We assume the values of all the grid points inside \( \Omega \) have been updated from time level \( t_{n-1} \) to time level \( t_n \). For a ghost point \( P = (x_i, y_j) \), we find a point \( P_0 = (x_0, y_0) = x_0 \) on the boundary \( \Gamma = \partial \Omega \) such that the normal \( n(x_0) \) at \( P_0 \) goes through \( P \). The sign of the normal \( n(x_0) \) is chosen in such a way that it is positive if it points to the exterior of \( \Omega \). The point \( P_0 \) and the normal \( n(x_0) \) can be obtained analytically, since we assume we have an explicit expression for the geometry of \( \Gamma \). We set up a local coordinate system at \( P_0 \) by

\[
\begin{pmatrix}
\hat{x} \\
\hat{y}
\end{pmatrix} = \begin{pmatrix}
\cos \theta & \sin \theta \\
-\sin \theta & \cos \theta
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix} = \mathbf{T}
\begin{pmatrix}
x \\
y
\end{pmatrix},
\]

where \( \theta \) is the angle between the normal \( n(x_0) \) and the \( x \)-axis and \( \mathbf{T} \) is a rotational matrix. The \( \hat{x} \)-axis then points in the same direction as \( n(x_0) \) and the \( \hat{y} \)-axis points in the tangential direction, see Figure 2.

In this local coordinate system, the Euler equations (11) are written as

\[
\dot{\hat{U}} + F(\hat{U})_{\hat{x}} + G(\hat{U})_{\hat{y}} = 0,
\]

where

\[
\hat{U} = \begin{pmatrix}
\hat{U}_1 \\
\hat{U}_2 \\
\hat{U}_3 \\
\hat{U}_4
\end{pmatrix} = \begin{pmatrix}
\rho \\
\rho \hat{u} \\
\rho \hat{v} \\
E
\end{pmatrix}, \quad \begin{pmatrix}
\hat{u} \\
\hat{v}
\end{pmatrix} = \mathbf{T}
\begin{pmatrix}
u \\
v
\end{pmatrix}.
\]
Figure 2: The local coordinate system (13) used in the ILW procedure for 2D Euler equations. For static geometry, $t_n$ dependence can be suppressed.

For a fifth order boundary treatment, the value of the ghost point $P$ is imposed by the Taylor expansion

$$
(\hat{U}_m)_{ij} = \sum_{k=0}^{4} \frac{\Delta^k}{k!} \tilde{U}_m^{(k)}, \quad m = 1, \ldots, 4,
$$

where $\Delta$ is the $\hat{x}$-coordinate of $P$ and $\tilde{U}_m^{(k)}$ is a $(5-k)$th order approximation of the normal derivative $\left. \frac{\partial^{k} \hat{U}_m}{\partial x^{k}} \right|_{(x,y)=x_0, t=t_n}$. We assume $\hat{U}_0$ is the value of a grid point nearest to $P_0$ among all the grid points inside $\Omega$. We denote the Jacobian matrix of the normal flux by

$$
A_{\perp}(\hat{U}_0) = \left. \frac{\partial \mathbf{F}(\hat{U})}{\partial \hat{U}} \right|_{\hat{U}=\hat{U}_0}.
$$

$A_{\perp}(\hat{U}_0)$ has four eigenvalues $\lambda_1 = \hat{u}_0 - c_0$, $\lambda_2 = \lambda_3 = \hat{u}_0$, $\lambda_4 = \hat{u}_0 + c_0$ and a complete set of left eigenvectors $l_m(\hat{U}_0)$, $m = 1, \ldots, 4$, which forms a left eigenmatrix

$$
\mathbf{L}(\hat{U}_0) = \begin{pmatrix}
    l_1(\hat{U}_0) \\
    l_2(\hat{U}_0) \\
    l_3(\hat{U}_0) \\
    l_4(\hat{U}_0)
\end{pmatrix} =
\begin{pmatrix}
    l_{1,1} & l_{1,2} & l_{1,3} & l_{1,4} \\
    l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\
    l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\
    l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4}
\end{pmatrix}.
$$

For definiteness, we assume $\lambda_1 < 0$ and $\lambda_4 > \lambda_2 = \lambda_3 > 0$. Thus one boundary condition is needed at $P_0$ according to 1D well-posedness theory. For example, the normal momentum is prescribed $\hat{U}_2(x_0, t) = g_2(t)$. 

15
The local characteristic variables $V_m$ at grid points near $P_0$ are defined by

$$(V_m)_{\mu,\nu} = l_m(\hat{U}_0)\hat{U}_{\mu,\nu}, \quad m = 1,\ldots, 4, \quad (x_\mu, y_\nu) \in \mathcal{E}_{i,j},$$

where $\mathcal{E}_{i,j} \subset \Omega$ is a set of grid points used for the fifth order 2D WENO extrapolation (Tan et al., 2012). We extrapolate $(V_m)_{\mu,\nu}$ to $P_0$ and denote the extrapolated $k$th order $\hat{x}$-derivative of $V_m$ by $V_m^{(k)}$, $k = 0,\ldots, 4$.

We solve $\hat{U}_m^{(0)}$ by a linear system of equations

$$\begin{pmatrix}
0 & 1 & 0 & 0 \\
l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\
l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\
l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4}
\end{pmatrix}
\begin{pmatrix}
\hat{U}_1^{(0)} \\
\hat{U}_2^{(0)} \\
\hat{U}_3^{(0)} \\
\hat{U}_4^{(0)}
\end{pmatrix}
= \begin{pmatrix}
g_2(t_n) \\
V_2^{(0)} \\
V_3^{(0)} \\
V_4^{(0)}
\end{pmatrix}. \tag{17}
$$

Here the first equation is the prescribed boundary condition. The other equations represent extrapolation of the three outgoing characteristic variables $V_m$, $m = 2,\ldots, 4$. Next, we use the ILW procedure for $\hat{U}_2$. The second equation of (14) gives us

$$\frac{\partial \hat{U}_2}{\partial t} = -\left(\frac{\gamma - 3}{2}\hat{U}_1^2 + \frac{\gamma - 1}{2} \hat{U}_3^2\right) \frac{\partial \hat{U}_1}{\partial \hat{x}} - (3 - \gamma) \frac{\hat{U}_2 \partial \hat{U}_2}{\hat{U}_1} \frac{\partial \hat{U}_1}{\partial \hat{x}}$$

$$+ (\gamma - 1) \frac{\hat{U}_3 \partial \hat{U}_3}{\hat{U}_1} - (\gamma - 1) \frac{\partial \hat{U}_4}{\partial \hat{y}} - \frac{\partial}{\partial \hat{y}} \left( \frac{\hat{U}_2 \hat{U}_3}{\hat{U}_1} \right).$$

At the boundary, the left-hand side of the above equation is the known function $g'_2(t)$. The tangential derivative on the right-hand side can be computed by numerical differentiation, since we have obtained $\hat{U}_m^{(0)}$ of all the ghost points. Thus $\hat{U}_m^{(1)}$ can be solved by the linear system

$$A^{(0)} \begin{pmatrix}
\hat{U}_1^{(1)} \\
\hat{U}_2^{(1)} \\
\hat{U}_3^{(1)} \\
\hat{U}_4^{(1)}
\end{pmatrix}
= \begin{pmatrix}
-g'_2(t_n) - \frac{\partial}{\partial \hat{y}} \left( \frac{\hat{U}_2^{(0)} \hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} \right) \\
V_2^{(1)} \\
V_3^{(1)} \\
V_4^{(1)}
\end{pmatrix}, \tag{18}
$$

where

$$A^{(0)} = \begin{pmatrix}
\frac{\gamma - 3}{2} \left( \frac{\hat{U}_1^{(0)}}{\hat{U}_1^{(0)}} \right)^2 + \frac{\gamma - 1}{2} \left( \frac{\hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} \right)^2 & (3 - \gamma) \frac{\hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} & (1 - \gamma) \frac{\hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} & \gamma - 1 \\
l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\
l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\
l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4}
\end{pmatrix}. $$
There are two ways to obtain the higher order derivatives $\hat{U}_m^{(k)}$, $k = 2, 3, 4$: the ILW procedure and extrapolation. The first option allows the same CFL number as the periodic boundary case and thus allows the largest possible time step size (Li et al., 2016). The ILW procedure however involves quite heavy algebra. The other option is easy to implement, but it would lead to a more restrictive CFL condition. Here we choose extrapolation. Namely, $\hat{U}_m^{(k)}$, $k = 2, 3, 4$, are solved by extrapolation equations

$$L(\hat{U}_0) \begin{pmatrix} \hat{U}_1^{(k)} \\ \hat{U}_2^{(k)} \\ \hat{U}_3^{(k)} \\ \hat{U}_4^{(k)} \end{pmatrix} = \begin{pmatrix} V_1^{(k)} \\ V_2^{(k)} \\ V_3^{(k)} \\ V_4^{(k)} \end{pmatrix}.$$  

(19)

One of the most common boundary conditions in computational fluid mechanics is the no-penetration boundary condition at rigid walls: $\hat{u} = 0$ or $\hat{U}_2 = 0$. In this case, the eigenvalues $\lambda_1 \approx -c_0 < 0$, $\lambda_4 \approx c_0 > 0$ and $\lambda_2 = \lambda_3 \approx 0$. Since only one boundary condition is prescribed, we consider $V_m$, $m = 2, \ldots, 4$, to be outgoing and $V_1$ to be ingoing, which falls into the same case as discussed above. (17) gives us $\hat{U}_2^{(0)} = 0$. Then the first equation of (18) reduces to

$$\frac{\gamma - 1}{2} \left( \frac{\hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} \right)^2 \hat{U}_1^{(1)} + (1 - \gamma) \frac{\hat{U}_3^{(0)}}{\hat{U}_1^{(0)}} \hat{U}_3^{(1)} + (\gamma - 1) \hat{U}_4^{(1)} = \frac{\left( \hat{U}_3^{(0)} \right)^2}{R \hat{U}_1^{(0)}},$$  

(20)

where $R$ is the radius of curvature of $\partial \Omega$ at $P_0$. Notice that there is no tangential derivative in (20). Therefore, we do not need to do any numerical differentiation, which further simplifies the implementation.

(20) can also be derived by considering primitive variables, i.e., $\rho, \hat{u}, \hat{v}$, and $p$, in the ILW procedure. At the rigid walls, we obtain

$$\frac{\partial p}{\partial \hat{x}} = \rho \frac{\hat{v}^2}{R},$$

which is equivalent to (20) because of the equation of state (12). In fact, it is sometimes more convenient to use primitive variables than to use conservative variables $U$. For prob-
lems involving moving geometry, we can only use primitive variables because the boundary condition is prescribed in normal velocity. See Section 4 for details.

We now summarize our fifth order boundary treatment for the 2D problem (11). We assume the values of all the grid points inside Ω have been updated from time level \( t_{n-1} \) to time level \( t_n \). Our goal is to impose the value of \((\hat{U}_m)_{i,j}, m = 1, \ldots, 4\), for each ghost point \((x_i, y_j)\).

1. For each ghost point \((x_i, y_j)\), we do the following three steps:

   - Decide the local coordinate system (13). Compute the eigenvalues \( \lambda_m(\hat{U}_0) \) and left eigenvectors \( l_m(\hat{U}_0) \) of the Jacobian matrix \( A_\perp(\hat{U}_0) \) for \( m = 1, \ldots, 4 \). Decide the prescribed inflow boundary conditions \( g_m(t) \) according to the signs of \( \lambda_m(\hat{U}_0) \).
   
   - Form the local characteristic variables \((V_m)_{\mu,\nu}, (x_\mu, y_\nu) \in E_{i,j}\) as in (16). Extrapolate \((V_m)_{\mu,\nu}\) to the boundary to obtain \( V_m^{*}(k), k = 0, \ldots, 4\), with the fifth order 2D WENO extrapolation (Tan et al., 2012).
   
   - Solve for \( \hat{U}^{*}(0)_m, m = 1, \ldots, 4, \) by the prescribed boundary conditions and extrapolated values \( V_m^{*}(0) \), such as (17).

2. For each ghost point \((x_i, y_j)\), use the ILW procedure to write the first derivative of \( g_m(t) \) as a linear combination of first normal derivatives plus tangential derivatives. Together with the extrapolation equations, form a linear system with \( \hat{U}_m^{*}(1) \) as unknowns, such as (18). Solve for \( \hat{U}_m^{*}(1), m = 1, \ldots, 4, \). For \( k = 2, 3, 4, \) solve for \( \hat{U}_m^{*}(k) \) by extrapolation equations (19) where the ILW procedure is not used.

3. Impose the values of the ghost points by the Taylor expansion (15).

If no-penetration boundary condition is considered at rigid walls, then the first equation of (18) is replaced by (20) in Step 2 with other steps unchanged. An alternative is to use primitive variables instead of conservative variables.
4 Moving boundary treatment for compressible inviscid flows

The advantage of using fixed Cartesian mesh may best be shown by problems involving complex moving geometry. We consider Euler equations in moving geometry, i.e., the domain \( \Omega(t) \) varies with time \( t \). To describe the boundary conditions, we let \( X_b(a, t) \) represent the position vector (in Eulerian coordinates) of a point \( a \) on \( \Gamma(t) \). Here \( a \) is the Lagrangian coordinate of the point determined by the condition \( X_b(a, 0) = a \). We mainly consider rigid bodies moving at a prescribed motion. Namely, \( X_b(a, t) \) and thus \( \Omega(t) \) are explicitly given. The no-penetration boundary condition for inviscid flows is then

\[
\mathbf{u}(x, t) \cdot \mathbf{n}(x, t) = V_b(t) \cdot \mathbf{n}(x, t), \quad \text{for all } x = X_b(a, t) \in \Gamma(t),
\]

where \( V_b(t) = \frac{\partial X_b}{\partial t} \) is the prescribed velocity. Notice that \( V_b(t) \) is independent of \( a \) for rigid bodies. The normal \( \mathbf{n}(x, t) \) is defined as the same way as in Figure 2. If the motion of a rigid body is induced by the fluid, the acceleration can be expressed as

\[
\frac{\partial^2 X_b}{\partial t^2} = -\frac{1}{M_b} \int_{\Gamma(t)} p \mathbf{n} ds,
\]

where \( M_b \) is the rigid body mass. Although \( X_b(a, t) \) is not explicitly given in this case, we can obtain it at each time level by integrating (22) in time.

There are two underlying issues with the moving boundary problem. First, the no-penetration condition (21) is prescribed in the Lagrangian specification of normal velocity. Thus in the ILW procedure we should use material derivatives of primitive variables, i.e., \( \rho, u, v, p \), instead of Eulerian time derivatives of conservative variable \( \mathbf{U} \). Secondly, in expansion flows there may be grid points which are outside \( \Omega(t_{n-1}) \) at the previous time level \( t_{n-1} \) but enter \( \Omega(t_n) \) at the current time level \( t_n \). Such grid points are called newly emerging points. The newly emerging points do not have any value at time level \( t_n \). To update their values to time level \( t_{n+1} \), we not only need to construct their values at time level \( t_n \), but also need one extra ghost point in each direction if we assume \( \Gamma(t) \) travels a distance of at most \( h \) in
each direction from \( t_n \) to \( t_{n+1} \), i.e.,
\[
\Delta t < \frac{h}{\max_{t \in [t_n, t_{n+1}]} \| V_b(t) \|_{\infty}}. \tag{23}
\]

See Figure 3 for a demonstration of newly emerging points and ghost points. If the same method is used to construct values of ghost points and newly emerging points, there is actually no need to distinguish them in implementation. We only need to construct values of four ghost points (instead of three ghost points for the static boundary problem) in each direction at time level \( t_n \).

![Figure 3: Newly emerging points and ghost points at time level \( t_n \) when the boundary is moving.](image)

We start to describe a fifth order boundary treatment for the no-penetration boundary condition (21) in which \( X_b(a, t) \) and thus \( V_b(t) \) are prescribed. We assume the values of all the grid points inside \( \Omega(t_{n-1}) \) have been updated by the interior scheme from time level \( t_{n-1} \) to time level \( t_n \). At time level \( t_n \), we set up the local coordinate system (13) with origin \( P_0 = x_0 \) for each ghost point \( P = (x_i, y_j) \). In this local coordinate system, the Euler equations are written in terms of primitive variable as
\[
\frac{\partial \hat{W}}{\partial t} + A(\hat{W}) \frac{\partial \hat{W}}{\partial \hat{x}} + B(\hat{W}) \frac{\partial \hat{W}}{\partial \hat{y}} = 0, \tag{24}
\]
where

\[
\mathbf{\hat{W}} = \begin{pmatrix} \hat{W}_1 \\ \hat{W}_2 \\ \hat{W}_3 \\ \hat{W}_4 \end{pmatrix} = \begin{pmatrix} \rho \\ \hat{u} \\ \hat{v} \\ \rho \end{pmatrix}, \quad \mathbf{A}(\mathbf{\hat{W}}) = \begin{pmatrix} \hat{u} & \rho & 0 & 0 \\ 0 & \hat{u} & 0 & \frac{1}{\rho} \\ 0 & 0 & \hat{u} & 0 \\ 0 & \rho \epsilon^2 & 0 & \hat{u} \end{pmatrix}, \quad \mathbf{B}(\mathbf{\hat{W}}) = \begin{pmatrix} \hat{v} & 0 & \rho & 0 \\ 0 & \hat{v} & 0 & 0 \\ 0 & 0 & \hat{v} & \frac{1}{\rho} \\ 0 & 0 & \rho \epsilon^2 & \hat{v} \end{pmatrix}.
\]

Our ILW procedure is performed using (24).

For a fifth order boundary treatment, the value of the ghost point \( P = (x_i, y_j) \) is imposed by the Taylor expansion

\[
(\hat{W}_m)_{ij} = \sum_{k=0}^{4} \frac{\Delta^k}{k!} \hat{W}^{(k)}_m, \quad m = 1, \ldots, 4,
\]

where \( \Delta \) is the \( \hat{x} \)-coordinate of \( P \) and \( \hat{W}^{(k)}_m \) is a \((5 - k)\)th order approximation of the normal derivative \( \frac{\partial \hat{W}_m}{\partial x} \bigg|_{(x, y) = x_0, \ t = t_n} \). We assume \( \hat{W}_0 \) is the value of a grid point nearest to \( P_0 \) among all the grid points inside \( \Omega(t_{n-1}) \). \( \mathbf{A}(\hat{W}_0) \) has four eigenvalues \( \lambda_1 = \hat{u}_0 - c_0, \lambda_2 = \lambda_3 = \hat{u}_0, \lambda_4 = \hat{u}_0 + c_0 \) and a complete set of left eigenvectors \( \mathbf{l}_m(\hat{W}_0), m = 1, \ldots, 4 \), which forms a left eigenmatrix

\[
\mathbf{L}(\hat{W}_0) = \begin{pmatrix} \mathbf{l}_1(\hat{W}_0) \\ \mathbf{l}_2(\hat{W}_0) \\ \mathbf{l}_3(\hat{W}_0) \\ \mathbf{l}_4(\hat{W}_0) \end{pmatrix} = \begin{pmatrix} l_{1,1} & l_{1,2} & l_{1,3} & l_{1,4} \\ l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\ l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\ l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4} \end{pmatrix}.
\]

The characteristic variables \( V_m, m = 1, \ldots, 4 \), at grid points near \( P_0 \) can be defined by

\[
(V_m)_{\mu, \nu} = \mathbf{l}_m(\hat{W}_0) \hat{W}_{\mu, \nu}, \quad m = 1, \ldots, 4, \quad (x_\mu, y_\nu) \in \mathcal{E}_{ij},
\]

where \( \mathcal{E}_{ij} \subset \Omega(t_{n-1}) \) is a set of grid points used for extrapolation. We extrapolate \( (V_m)_{\mu, \nu} \) to \( P_0 \) and denote the extrapolated \( k \)th order \( \hat{x} \)-derivative of \( V_m \) by \( V^{(k)}_m \), \( k = 0, \ldots, 4 \). The constant term \( \hat{W}^{(0)}_m, m = 1, \ldots, 4 \), is solved by a linear system of equations

\[
\begin{pmatrix} 0 & 1 & 0 & 0 \\ l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\ l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\ l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4} \end{pmatrix} \begin{pmatrix} \hat{W}^{(0)}_1 \\ \hat{W}^{(0)}_2 \\ \hat{W}^{(0)}_3 \\ \hat{W}^{(0)}_4 \end{pmatrix} = \begin{pmatrix} V^{(0)}_b(t_n) \cdot \mathbf{n}(x_0, t_n) \\ V^{(0)}_2 \\ V^{(0)}_3 \\ V^{(0)}_4 \end{pmatrix}.
\]

Next, we take the first material derivative \( \frac{D}{Dt} = \frac{\partial}{\partial t} + \hat{u} \frac{\partial}{\partial x} + \hat{v} \frac{\partial}{\partial y} \) of (21) and obtain

\[
\frac{D\hat{u}}{Dt} + \hat{u} \cdot \frac{D\mathbf{n}}{Dt} = \frac{d}{dt}(\mathbf{V}_b \cdot \mathbf{n}),
\]
where \( \mathbf{u} = (\hat{u}, \hat{v})^T \). Converting the material derivative \( \frac{D\mathbf{u}}{Dt} \) to spatial derivatives by the second equation of (24), we have

\[
\frac{\partial p}{\partial \hat{x}} = \rho \left[ \mathbf{u} \cdot \frac{D\mathbf{n}}{Dt} - \frac{d}{dt} (\mathbf{V}_b \cdot \mathbf{n}) \right].
\]

The right-hand side of the above equation is already known if evaluated at \( P_0 \). As a result, \( \hat{W}^{(1)}_m, m = 1, \ldots, 4 \), can be solved by

\[
\begin{pmatrix}
0 & 0 & 0 & 1 \\
l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} \\
l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} \\
l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4}
\end{pmatrix}
\begin{pmatrix}
\hat{W}^{(1)}_1 \\
\hat{W}^{(1)}_2 \\
\hat{W}^{(1)}_3 \\
\hat{W}^{(1)}_4
\end{pmatrix}
= \mathbf{b},
\]

where

\[
\mathbf{b} = \left. \left( \hat{W}^{(0)}_1 \left[ \begin{array}{c}
\hat{W}^{(0)}_2 \\
\hat{W}^{(0)}_3
\end{array} \right] \right)^T \cdot \frac{D\mathbf{n}}{Dt} - \frac{d}{dt} (\mathbf{V}_b \cdot \mathbf{n}) \right|_{(x,y)=x_0, t=t_n}.
\]

Higher order spatial derivatives \( \hat{W}^{(k)}_m, k = 2, 3, 4 \), can be obtained by extrapolation

\[
\mathbf{L}(\hat{\mathbf{W}}_0) \begin{pmatrix}
\hat{W}^{(k)}_1 \\
\hat{W}^{(k)}_2 \\
\hat{W}^{(k)}_3 \\
\hat{W}^{(k)}_4
\end{pmatrix} = \begin{pmatrix}
V^{(k)}_1 \\
V^{(k)}_2 \\
V^{(k)}_3 \\
V^{(k)}_4
\end{pmatrix}.
\]

We now summarize our fifth order boundary treatment for the boundary condition (21) in 2D with a prescribed boundary motion. Our goal is to impose the value of \( (\hat{W}_m)_{i,j}, m = 1, \ldots, 4 \), for each ghost point \((x_i, y_j)\) at time level \( t = t_n \).

1. Decide the local coordinate system (13). Do a local characteristic decomposition of the Euler equations (24). Form the characteristic variables \( (\mathbf{V}_m)_{\mu,\nu}, m = 1, \ldots, 4, (\mu, \nu) \in \mathcal{E}_{i,j} \), as in (26). Extrapolate \( (\mathbf{V}_m)_{\mu,\nu} \) to the boundary to obtain \( V^{(k)}_m, k = 0, \ldots, 4 \), with WENO extrapolation. Solve for \( \hat{W}^{(0)}_m, m = 1, \ldots, 4 \), in (27).

2. Use the ILW procedure with material derivatives and the extrapolation equations to form linear system (28). Solve for \( \hat{W}^{(1)}_m, m = 1, \ldots, 4 \).
3. For \( k = 2, 3, 4 \), solve for \( \hat{W}^{(k)}_m \), \( m = 1, \ldots, 4 \), in (29).

4. Impose the values of the ghost points by the Taylor expansion (25).

If the motion of a rigid body is not prescribed but induced by the fluid, our algorithm should be adjusted as follows. Before Step 2, we compute \( \frac{dV_b}{dt} = \frac{\partial^2 X_b}{\partial t^2} \) by (22), since we have obtained pressure \( p \) at \( P_0 \) on \( \Gamma(t_n) \) for all the ghost points. The integral in (22) can be calculated by the trapezoidal rule, for example. Notice that integrating (22) in time by the same Runge-Kutta method, we can obtain \( X_b(a, t_{n+1}) \) and \( V_b(t_{n+1}) \) so that our algorithm can be continued at next time level \( t_{n+1} \).

We finally remark that the boundary treatment for moving geometry can clearly be applied to static geometry as well. This provides an alternative to the method developed in Section 3.

5 Numerical results

In this section we present several numerical results obtained using the fifth order ILW procedure boundary treatment.

**Example 1** We test the double Mach reflection problem (Woodward and Colella, 1984) which involves a rigid wall not aligned with the grid lines. This problem is initialized by sending a horizontally moving shock into a rigid wall inclined by a 30° angle. In order to impose the no-penetration boundary condition by the reflection technique, people usually solve an equivalent problem in which the wall is horizontal and the shock is inclined to the wall at 60° angle (e.g., Jiang and Shu, 1996; Shi et al., 2003). We consider the solution of the equivalent problem as our reference solution. Another way to apply the reflection technique is to use a multidomain WENO method (Sebastian and Shu, 2003).

We are able to solve the original problem on a uniform mesh with our fifth order boundary treatment in conservative variables. The computational domain is sketched in Figure 4(a), together with some of the grid points near the wall indicating that the wall is not aligned
with the grid lines. Initially a right-moving Mach 10 shock is positioned at (0, 0) making an angle of $90^\circ$ with the $x$-axis. At $y = 0$, the exact postshock condition is imposed. At the top boundary, the flow values are set to describe the exact motion of the Mach 10 shock. Supersonic inflow and outflow boundary conditions are used at the left and right boundary respectively. Figure 4(b) shows the density contour with $h = 1/320$ at $t = 0.2$. A zoomed-in region near the double Mach stem is shown in Figure 5(a). The region is rotated and translated for the ease of comparison. In Figure 5(b), we show the reference solution on a mesh with comparable size. Figures 5(c) and 5(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 2 This is an example involving a curved wall. The problem is initialized by a Mach 3 flow moving toward a circular cylinder of unit radius positioned at the origin on a $x$-$y$ plane. In order to impose the no-penetration 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 (Jiang and Shu, 1996). Using our boundary treatment, we are able to solve this problem directly in the physical domain, which is shown in Figure 6, together with some of the grid points near the cylinder which indicate boundary cuts the grid in an arbitrary fashion. The 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. Supersonic inflow boundary condition is used at the left boundary $x = -3$; supersonic outflow boundary condition is used at the top boundary $y = 6$ and at the right boundary $x = 0$. Our fifth order boundary treatment in primitive variables is applied at the surface of the cylinder.

The pressure contour at steady state is shown in Figure 7 with different mesh sizes. The bow shock is well-captured by our method. For a more quantitative verification, we take advantage of the entropy along the surface, which can be computed analytically by using the
Figure 4: Top: The computational domain (solid line) of the double Mach reflection problem in Example 1. The dashed line indicates the computational domain used in the equivalent problem (e.g., Jiang and Shu, 1996; Shi et al., 2003). The square points indicate some of the grid points near the wall. Illustrative graph, not to scale. Bottom: Density contour of double Mach reflection, 30 contour lines from 1.731 to 20.92. $h = 1/320$. 

(a) computational domain

(b) density contour
Figure 5: Density contours of the double Mach reflection problem in Example 1, 30 contour lines 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.
Rankine-Hugoniot conditions for the streamline normal to the bow shock at $y = 0$. Since there is usually no grid point located on the surface, we compute the entropy errors of the state $\hat{U}_m^{(0)}$, $m = 1, \ldots, 4$, which is the constant term in the Taylor expansion (15), for all the ghost points. We can see superlinear convergence rates in Table 2. In other words, although the accuracy of our high order boundary treatment degenerates to first order near the shock, the accuracy in the smooth part of the solution is expected to be higher than first order.

Figure 6: Physical domain of the flow past a cylinder problem in Example 2. The square points indicate some of the grid points near the cylinder. Illustrative sketch, not to scale.

Figure 7: Pressure contours of the flow past a cylinder problem in Example 2, 20 contour lines from 2 to 15.
Table 2: $L^\infty$ entropy errors on the surface of the cylinder and rates of convergence for the flow past a cylinder problem in Example 2.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$L^\infty$ error</th>
<th>Order</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/20</td>
<td>1.68E-02</td>
<td></td>
</tr>
<tr>
<td>1/40</td>
<td>7.06E-03</td>
<td>1.25</td>
</tr>
<tr>
<td>1/80</td>
<td>1.88E-03</td>
<td>1.91</td>
</tr>
<tr>
<td>1/160</td>
<td>7.50E-04</td>
<td>1.33</td>
</tr>
</tbody>
</table>

Example 3 The last example shows that our high order method can also treat a rigid body whose motion is induced by the fluid. We test the so-called cylinder lift-off problem which is first introduced by Falcovitz et al. (1997) and discussed in later work (Arienti et al., 2003; Forrer and Berger, 1999; Hu et al., 2006; Shyue, 2008). In this problem, a rigid cylinder initially resting on the floor of a 2D channel is driven and lifted by a strong shock. The computational domain is $[0, 1] \times [0, 0.2]$. A rigid cylinder with radius 0.05 and density 10.77 is initially centered at $(0.15, 0.05)$. A Mach 3 shock starts at $x = 0.08$ moving towards the cylinder. The density and pressure of the resting gas are $\rho = 1.4$ and $p = 1.0$ respectively. The top and bottom of the domain are rigid walls. The left boundary is set to the post-shock state and the right boundary is supersonic outflow.

We use our fifth order boundary treatment at the surface of the moving cylinder and the reflection technique at the top and bottom walls. Since the cylinder initially rests exactly on the floor, a stencil for high order extrapolation may be too wide to be contained in the computational domain. We have to use low order extrapolation in this situation and turn to high order extrapolation otherwise. We list the center of the cylinder at two fixed times for different meshes in Table 3. The results imply convergence. We plot pressure contours at $t = 0.1641$ and $t = 0.30085$ in Figure 8 and Figure 9 respectively. The flow structures agree with those obtained by Arienti et al. (2003), Hu et al. (2006), and Shyue (2008).
Table 3: Center of the cylinder in the cylinder lift-off problem in Example 3.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$t = 0.1641$</th>
<th>$t = 0.30085$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$-coordinate</td>
<td>$y$-coordinate</td>
</tr>
<tr>
<td>1/320</td>
<td>3.5878E-01</td>
<td>8.4001E-02</td>
</tr>
<tr>
<td>1/640</td>
<td>3.5542E-01</td>
<td>8.3778E-02</td>
</tr>
<tr>
<td>1/1280</td>
<td>3.5434E-01</td>
<td>8.3827E-02</td>
</tr>
<tr>
<td>1/2560</td>
<td>3.5422E-01</td>
<td>8.4140E-02</td>
</tr>
</tbody>
</table>

Figure 8: Pressure contours of the cylinder lift-off problem in Example 3, 53 contours from 2 to 28. $t = 0.1641$. 

(a) $h = 1/640$

(b) $h = 1/1280$
Figure 9: Pressure contours of the cylinder lift-off problem in Example 3, 53 contours from 2 to 28. \( t = 0.30085 \).

## 6 Conclusions and future work

We discuss a numerical boundary condition of hyperbolic equations on fixed Cartesian grids. The grid lines can cut through the physical domain in any possible way, which poses a significant challenge for accuracy and stability of boundary conditions. We impose the inflow boundary condition using the inverse Lax-Wendroff procedure in which spatial derivatives are obtained from the partial differential equations. The outflow boundary condition is treated by WENO extrapolation. Our method is high order accurate (up to fifth order is discussed in this article), stable, and easy to implement. We have successfully applied the method to the simulation of interactions between compressible inviscid flows and rigid (static or moving) bodies with complex geometry.

The challenge of high order boundary treatment is not limited to finite difference schemes. Even for finite element type methods, difficulties sometimes arise if unstructured, straight-sided meshes are used to fit curved, static geometry (e.g., Bassi and Rebay, 1997). Accurate implementations of boundary conditions for discontinuous Galerkin (DG) methods on such meshes have been developed (Krivodonova and Berger, 2006; Zhang, 2016). We will try to
extend our boundary treatment to DG methods on rectangular meshes in our future work.

We have only considered the interaction between compressible inviscid flows and rigid bodies without deformation. In general fluid-structure interaction problems, the fluids can be viscous and the geometrically complicated structures are considered to be elastic or plastic. A closely related problem is the multi-fluid problem that involves internal boundaries separating different fluids. Low order coupling methods have been developed for a coupling of compressible inviscid flows and elastic solids (Arienti et al., 2003), and for a multi-fluid coupling (Hu et al., 2006). Our high order boundary treatment looks promising for more accurate couplings in both types of problems.

References


