

RESEARCH

Open Access



Numerical simulation of a complex moving rigid body under the impingement of a shock wave in 3D

Shihao Liu¹, Ziqiang Cheng², Yan Jiang^{1*}, Jianfang Lu³, Mengping Zhang¹ and Shuhai Zhang⁴

*Correspondence:

jiangy@ustc.edu.cn

¹School of Mathematical Sciences, University of Science and Technology of China, 230026 Hefei, Anhui, China

Full list of author information is available at the end of the article

Abstract

In this paper, we take a numerical simulation of a complex moving rigid body under the impingement of a shock wave in three-dimensional space. Both compressible inviscid fluid and viscous fluid are considered with suitable boundary conditions. We develop a high order numerical boundary treatment for the complex moving geometries based on finite difference methods on fixed Cartesian meshes. The method is an extension of the inverse Lax-Wendroff (ILW) procedure in our works (Cheng et al., *Appl Math Mech (Engl Ed)* 42: 841–854, 2021; Liu et al.) for 2D problems. Different from the 2D case, the local coordinate rotation in 3D required in the ILW procedure is not unique. We give a theoretical analysis to show that the boundary treatment is independent of the choice of the rotation, ensuring the method is feasible and valid. Both translation and rotation of the body are taken into account in this paper. In particular, we reformulate the material derivative for inviscid fluid on the moving boundary with no-penetration condition, which plays a key role in the proposed algorithm. Numerical simulations on the cylinder and sphere are given, demonstrating the good performance of our numerical boundary treatments.

Keywords: Inverse Lax-Wendroff procedure, High order accuracy, Complex moving boundary treatment, Compressible inviscid fluid, Compressible viscous fluid, 3D

1 Introduction

In this paper, we design a high order boundary treatment combining high order finite difference scheme for both inviscid and viscous fluids in the 3D time-varying domain. The flow field is between two flat plates parallel to the xz -plane, and there is a rigid body inside it. At the initial moment, there is a plane shock wave perpendicular to the x -axis and moving to the positive direction of the x -axis, towards to the rigid body. Under the impingement of the flow, the rigid body starts to move in the region, then the area of the fluid varies with time, denoted by $\Omega(t)$.

In order to prescribe the boundary conditions on the surface of the rigid body, we take the notation $\Gamma(t)$ to represent the interface between the rigid body and fluid. Hence, we have the boundary of the domain

$$\partial\Omega(t) = \Gamma(t) \cup \Pi_x \cup \Pi_y \cup \Pi_z,$$

where, Π_x , Π_y and Π_z are the outer boundaries of the computational area along x -, y - and z -direction, parallel to the yz -, xz - and xy -plane, respectively. Inflow, outflow and reflection boundary conditions are imposed at the left boundary, the right boundary of Π_x and the two boundaries of Π_y , respectively. Notice that, although the physical problem we actually consider is infinite in the z -direction, the domain is truncated in numerical computation. Due to the fact that the rigid body moves in a confined space during a limited period, the area can be assumed to be large enough in the z -direction that the body and the reflective shock waves would not touch Π_z , and then we can impose the outflow boundary conditions on Π_z . Hence, the boundary treatment on Π_z will not affect the computational results of the internal flow field near the rigid boundary. The boundary conditions on the boundary $\Gamma(t)$ vary with the properties of the fluid. In this paper, we consider both the compressible inviscid fluid and viscous fluid with suitable boundary conditions.

The governing equations for compressible inviscid fluid and viscous fluid in the three-dimensions are written in the same form as follows:

$$\mathbf{W}_t + \mathbf{F}(\mathbf{W})_x + \mathbf{G}(\mathbf{W})_y + \mathbf{H}(\mathbf{W})_z = \frac{\delta}{Re} (\mathbf{S}_1(\mathbf{W})_x + \mathbf{S}_2(\mathbf{W})_y + \mathbf{S}_3(\mathbf{W})_z), \tag{1}$$

where $(x, y, z) \in \Omega(t)$, $\mathbf{W} = (\rho, \rho u, \rho v, \rho w, E)^T$ and

$$\mathbf{F}(\mathbf{W}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{pmatrix}, \quad \mathbf{G}(\mathbf{W}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E + p) \end{pmatrix}, \quad \mathbf{H}(\mathbf{W}) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E + p) \end{pmatrix}. \tag{2}$$

Here, ρ , u , v , w , p and E stand for density, velocity in x -, y - and z -directions, pressure and total energy per volume, respectively. On the right hand side of (1),

$$\mathbf{S}_1(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{31} \\ \sigma_1 \end{pmatrix}, \quad \mathbf{S}_2(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{32} \\ \sigma_2 \end{pmatrix}, \quad \mathbf{S}_3(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{13} \\ \tau_{23} \\ \tau_{33} \\ \sigma_3 \end{pmatrix},$$

with the components of the viscous stress tensor given by

$$\begin{aligned} \tau_{11} &= \frac{4}{3}u_x - \frac{2}{3}v_y - \frac{2}{3}w_z, & \tau_{22} &= \frac{4}{3}v_y - \frac{2}{3}u_x - \frac{2}{3}w_z, & \tau_{33} &= \frac{4}{3}w_z - \frac{2}{3}u_x - \frac{2}{3}v_y, \\ \tau_{12} &= \tau_{21} = v_x + u_y, & \tau_{13} &= \tau_{31} = u_z + w_x, & \tau_{23} &= \tau_{32} = w_y + v_z, \end{aligned} \tag{3}$$

and

$$\begin{aligned} \sigma_1 &= u \tau_{11} + v \tau_{12} + w \tau_{13} + \frac{(c^2)_x}{(\gamma - 1)Pr}, \\ \sigma_2 &= u \tau_{21} + v \tau_{22} + w \tau_{23} + \frac{(c^2)_y}{(\gamma - 1)Pr}, \\ \sigma_3 &= u \tau_{31} + v \tau_{32} + w \tau_{33} + \frac{(c^2)_z}{(\gamma - 1)Pr}. \end{aligned} \tag{4}$$

Here, Pr is the Prandtl number, Re is the Reynolds number, $T = p/\rho$ is the temperature, and $c = \sqrt{\gamma p/\rho}$ is the sound speed. An equation of state relating the pressure with other variables is given as

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2 + w^2) \right). \quad (5)$$

γ is the adiabatic constant, which equals to 1.4 for an ideal polytropic gas.

- **Compressible inviscid fluid.** In this case, $\delta = 0$ in the governing Eq. (1) and it becomes the three-dimensional *Euler equation*,

$$\mathbf{W}_t + \mathbf{F}(\mathbf{W})_x + \mathbf{G}(\mathbf{W})_y + \mathbf{H}(\mathbf{W})_z = \mathbf{0}, \quad (6)$$

The *no-penetration boundary conditions* are imposed on the boundary of the body surface $\Gamma(t)$:

$$\mathbf{u} \cdot \mathbf{n} = \mathbf{V}_b \cdot \mathbf{n}, \quad \forall \mathbf{x} \in \Gamma(t), \quad (7)$$

with $\mathbf{u} = (u, v, w)^T$. \mathbf{V}_b and \mathbf{n} are the velocity and the outward unit normal vector on $\Gamma(t)$, respectively.

- **Compressible viscous fluid.** In this case, $\delta = 1$ in the governing Eq. (1) and it is the three-dimensional *Navier-Stokes (NS) equation*,

$$\mathbf{W}_t + \mathbf{F}(\mathbf{W})_x + \mathbf{G}(\mathbf{W})_y + \mathbf{H}(\mathbf{W})_z = \frac{1}{Re} (\mathbf{S}_1(\mathbf{W})_x + \mathbf{S}_2(\mathbf{W})_y + \mathbf{S}_3(\mathbf{W})_z), \quad (8)$$

On $\Gamma(t)$, we consider the *isothermal no-slip wall boundary condition*:

$$\begin{cases} \mathbf{u} = \mathbf{V}_b \\ T = T_b \end{cases} \quad (9)$$

where $\mathbf{V}_b = (u_b, v_b, w_b)^T$ and T_b are the velocity and the temperature of the rigid body, respectively.

In this paper, we are trying to numerically solve the partial differential equations (PDEs) (6) and (8) with high order finite difference methods on the fixed Cartesian mesh. Despite the easy generation of the mesh, for such problems there are two main difficulties in numerical simulation:

1. A high order finite difference scheme often has a wide stencil, thus we have to evaluate the values at the ghost points near the boundary.
2. Another difficulty is that the Cartesian grids intersect with the physical boundary with arbitrary fashion. This often leads to the so-called “cut-cell” problem [1], e.g. in 1D case the first grid is very closed to the physical boundary. If the boundary treatment is not well designed, it may require the time step to be extremely small for the sake of stability and result in the poor computation efficiency. For the moving boundary problems, no matter how the mesh is generated initially, the “cut-cell” problem is unavoidable.

One of the commonly used methods is to generate a body-fitted grid. For simple and fixed boundaries, one may use the smooth mapping between the Cartesian mesh and curvilinear mesh, while this kind of treatment often needs to change the original equations. The new equations can be more complicated than the original equations and also bring additional computation. For the time-varying geometric domain, complex grid management is required at each time step and it may be extremely expensive. Another

approach is to use a Cartesian mesh, and take special treatment on the boundaries. There are several numerical boundary treatments listed as follows: embedded boundary method [2–7], immersed boundary method [8–12] and Inverse Lax-Wendroff (ILW) method [13–20], etc.

In this paper, we focus on the ILW boundary treatment for solving the moving boundary problems in 3D. The idea of ILW procedure comes from [21, 22], in which the authors simulated the traffic and pedestrian flow by solving an Eikonal equation at each time step. They used the PDE repeatedly to convert the normal derivative into the tangent derivative at the boundary, and the ghost point values are defined by a Taylor expansion along the normal direction. Enlightened by this approach, Tan and Shu extended it to hyperbolic conservation law equations in [17]. The main idea of the ILW method is to convert the spatial derivative into a time derivative through the PDE, which is in contrast to the traditional Lax-Wendroff scheme, in which the time derivative is converted into spatial derivative (this is exactly the meaning of “inverse”). Numerical results demonstrate that the ILW method can be applied to construct ghost point values when solving equations with high order finite difference methods on a Cartesian mesh in a computational domain with complex geometries. However, the original ILW method proposed in [17] has the following drawbacks:

1. Heavy algebra in the ILW procedure for high dimensional nonlinear system.
2. For the junction of the inflow and outflow boundary, i.e, the sonic points, the ILW method should be coupled with the least square method to ensure the stability of the scheme.
3. The numerical solution obtained by the ILW method cannot satisfy the conservation property while the exact solution of original PDE has this property.

There are several papers to address the issues above. In 2012 Tan et al. proposed the simplified ILW (SILW) method in [19], which greatly reduced the computation cost of the ILW method for solving the system. Ding et al. [14] redefined the concept of conservation for the finite difference scheme, and gave a new ILW method that satisfies the new conservation concept. Lu et al. [16] proposed an ILW method that can deal with the sonic points, which can avoid the situation of the extremely small denominator. Except for hyperbolic conservation laws, the ILW procedure can also be applied to other types of equations. In [23] Filbet and Yang extended the ILW method to handle the Boltzmann equation, and Li et al. studied the ILW procedure for the diffusion problems, and the authors in [15, 24, 25] extended the ILW method to handle the convection-diffusion equation. Stability analyses for both the ILW and the simplified ILW procedures were given in [24–27], in which the authors discussed various situations and gave us the guiding ideology for designing a high order and stable numerical boundary treatment. In particular, theoretical analysis indicates that the carefully designed (S)ILW boundary treatment can maintain stability with the same CFL number $(\lambda_{cfl})_{max}$ as in the periodic case, ignoring distances of the first grid point to the physical boundary. This means that the SILW method can overcome the “cut-cell” problem.

For moving boundary problems, in [18] Tan and Shu designed an ILW method for inviscid fluid with free-slip no-penetration moving boundary conditions, in which they only considered the translation of the moving boundary. Later, in [13] Cheng et al. reformulated the material derivative and considered both translation and rotation of

the body. Recently, [28] extended the moving boundary ILW method to deal with the convection-diffusion equations and simulate the interaction between shock waves and rigid bodies in viscous fluids. In particular, a unified algorithm was given for five cases: pure convection, convection-dominated, convection-diffusion, diffusion-dominated and pure diffusion cases. This paper is an extension of our works [13, 28], in which one and two dimensional moving boundary problems were considered. Now we extend these methods to handle three dimensional problems, with carefully designed numerical boundary conditions such that they can achieve third order of accuracy in smooth case, and no spurious oscillation when there is a shock near the surface. Particularly, in the ILW and SILW procedure, a local coordinate rotation on the boundary point is needed such that the \hat{x} -axis of the new coordinate is in the same direction with the outward normal on the boundary. Different from the two dimensional problem, this rotation is not unique. We prove that the rotation will not affect the results via ILW procedure. Hence, the schemes can work with a simple form of rotation. Since the conditions are defined on the moving boundaries, material derivatives are used in the ILW procedure instead of the Eulerian time derivatives. However, this definition of material derivatives for functions limited on the boundaries is not clear. In this paper, we will follow the idea in [13] and reformulate the material derivatives on the moving boundaries with no-penetration condition in 3D.

The organization of the paper is as follows. In Section 2, we introduce the spatial discretization and time evolution method for the Eqs. (6) and (8). In Section 3, we focus on the boundary treatment and present the ILW method in detail. The method of tracking the moving rigid body is discussed in Section 4. Numerical results are given in Section 5 and concluding remarks are given in Section 6.

2 Scheme formulation inside $\Omega(t)$

We assume that the computational domain $[x_l, x_r] \times [y_l, y_r] \times [z_l, z_r]$ is divided by a Cartesian mesh (x_i, y_j, z_k) , and the mesh sizes in each direction are uniform, denoted by Δx , Δy and Δz , respectively,

$$x_i = x_l + i \Delta x, \quad y_j = y_l + (j - 0.5) \Delta y, \quad z_k = z_l + k \Delta z. \quad (10)$$

Denote $\mathbf{W}_{i,j,k}(t)$ as the approximation to the point value $\mathbf{W}(x_i, y_j, z_k, t)$. We will discretize the Eqs. (6) and (8) in the method of lines framework, meaning that the spatial variable is first discretized, then the numerical solution is updated in time by coupling a suitable time integrator.

2.1 Spatial discretization

The semi-discrete approximation of the governing Eq. (1) is given as

$$\frac{d\mathbf{W}_{i,j,k}}{dt} + \frac{\hat{\mathbf{F}}_{i+1/2,j,k} - \hat{\mathbf{F}}_{i-1/2,j,k}}{\Delta x} + \frac{\hat{\mathbf{G}}_{i,j+1/2,k} - \hat{\mathbf{G}}_{i,j-1/2,k}}{\Delta y} + \frac{\hat{\mathbf{H}}_{i,j,k+1/2} - \hat{\mathbf{H}}_{i,j,k-1/2}}{\Delta z} = \mathbf{S}_{i,j,k}, \quad (11)$$

where $\hat{\mathbf{F}}_{i+1/2,j,k}$, $\hat{\mathbf{G}}_{i,j+1/2,k}$ and $\hat{\mathbf{H}}_{i,j,k+1/2}$ are numerical fluxes such that the flux difference approximates the derivative to r -th order accuracy,

$$\begin{aligned} \frac{\hat{\mathbf{F}}_{i+1/2,j,k} - \hat{\mathbf{F}}_{i-1/2,j,k}}{\Delta x} &= \mathbf{F}(\mathbf{W})_x|_{i,j,k} + \mathcal{O}(\Delta x^r), \\ \frac{\hat{\mathbf{G}}_{i,j+1/2,k} - \hat{\mathbf{G}}_{i,j-1/2,k}}{\Delta y} &= \mathbf{G}(\mathbf{W})_y|_{i,j,k} + \mathcal{O}(\Delta y^r), \\ \frac{\hat{\mathbf{H}}_{i,j,k+1/2} - \hat{\mathbf{H}}_{i,j,k-1/2}}{\Delta z} &= \mathbf{H}(\mathbf{W})_z|_{i,j,k} + \mathcal{O}(\Delta z^r). \end{aligned}$$

Those numerical fluxes can be obtained by any reasonable finite difference scheme, such as the finite difference (FD) WENO scheme [29], which can achieve high order accuracy in smooth region and avoid numerical oscillatory near discontinuities. Here, we use the third order FD WENO scheme to obtain the numerical fluxes, i.e. $r = 3$. Details are given in Appendix A.

For the Euler Eq. (6), $\mathbf{S}_{i,j,k} = \mathbf{0}$ on the right hand side. And for the NS Eq. (8), $\mathbf{S}_{i,j,k}$ is the approximation to the diffusion terms. Due to the effect of diffusion, we use central finite difference scheme without WENO to obtain $\mathbf{S}_{i,j,k}$. In particular, the five points central scheme will guarantee fourth order of accuracy, and the formulations for the (mixed) derivatives are given as follows:

$$\begin{aligned} \frac{\partial f}{\partial x}|_{i,j,k} &= \frac{1}{12\Delta x}(f_{i-2,j,k} - 8f_{i-1,j,k} + 8f_{i+1,j,k} - f_{i+2,j,k}), \\ \frac{\partial^2 f}{\partial x^2}|_{i,j,k} &= \frac{1}{12\Delta x^2}(-f_{i-2,j,k} + 16f_{i-1,j,k} - 30f_{i,j,k} + 16f_{i+1,j,k} - f_{i+2,j,k}), \\ \frac{\partial^2 f}{\partial x \partial y}|_{i,j,k} &= \frac{1}{144\Delta x \Delta y}((f_{i-2,j-2,k} - 8f_{i-1,j-2,k} + 8f_{i+1,j-2,k} - f_{i+2,j-2,k}) \\ &\quad - 8(f_{i-2,j-1,k} - 8f_{i-1,j-1,k} + 8f_{i+1,j-1,k} - f_{i+2,j-1,k}) \\ &\quad + 8(f_{i-2,j+1,k} - 8f_{i-1,j+1,k} + 8f_{i+1,j+1,k} - f_{i+2,j+1,k}) \\ &\quad - (f_{i-2,j+2,k} - 8f_{i-1,j+2,k} + 8f_{i+1,j+2,k} - f_{i+2,j+2,k})). \end{aligned} \tag{12}$$

2.2 Time discretization

The semi-discrete scheme (11) is equivalent to the first order ordinary differential equation (ODE) system

$$\frac{d\mathbf{W}_{i,j,k}}{dt} = \mathcal{L}(\{\mathbf{W}_{i,j,k}\}), \tag{13}$$

with $\mathcal{L}(\{\mathbf{W}_{i,j,k}\})$ containing all spatial discretization terms. For time discretization, we use the third-order total variation diminishing (TVD) Runge-Kutta method [30]:

$$\begin{aligned} \mathbf{W}_{i,j,k}^{(1)} &= \mathbf{W}_{i,j,k}^n + \Delta t \mathcal{L}(\{\mathbf{W}_{i,j,k}^n\}), \\ \mathbf{W}_{i,j,k}^{(2)} &= \frac{3}{4}\mathbf{W}_{i,j,k}^n + \frac{1}{4}\mathbf{W}_{i,j,k}^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(\{\mathbf{W}_{i,j,k}^{(1)}\}), \\ \mathbf{W}_{i,j,k}^{n+1} &= \frac{1}{3}\mathbf{W}_{i,j,k}^n + \frac{2}{3}\mathbf{W}_{i,j,k}^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(\{\mathbf{W}_{i,j,k}^{(2)}\}). \end{aligned} \tag{14}$$

In addition, a following constraint on time step Δt is imposed such that the boundary $\Gamma(t)$ can move at most one grid in each direction in a time step:

$$\Delta t \leq \min\left(\frac{\Delta x}{V_{b,x,max}}, \frac{\Delta y}{V_{b,y,max}}, \frac{\Delta z}{V_{b,z,max}}\right), \tag{15}$$

where $V_{b,x,max}$, $V_{b,y,max}$ and $V_{b,z,max}$ represent the maximum magnitude of the moving velocities in x -, y - and z - directions on the boundary $\Gamma(t)$.

3 Boundary treatments

In order to make the interior finite difference scheme work, we need to define enough ghost points near the boundaries. Near the outer boundaries $\Pi_x \cup \Pi_y \cup \Pi_z$, which consist of straight lines, ghost points are assigned according to the principles of inflow, outflow and reflection boundary conditions. While near the inner boundary $\Gamma(t)$, we will assign carefully designed values on ghost points. In addition, there is a kind of points called “newly emerging” points that deserve special attention, which are outside the computational domain $\Omega(t_{n-1})$ at the last time and entering $\Omega(t_n)$ at the current time due to the movement of the boundary. In [18], the authors pointed out that the newly emerging points do not need any special treatment, but just set at most one more ghost point in each direction. Then, we can use the same scheme as the interior points to update the values of these points at the last stage of (14). And that is exactly the reason why we give a constraint on the time step in (15).

Note that those ghost points may not lie on the grid. Hence, we follow the idea given in [17] that the method consists of the inverse Lax-Wendroff (ILW) type procedure for inflow boundary conditions and extrapolation for outflow boundary conditions. At the outflow boundary, WENO type extrapolation will be used in case of a shock close to the boundary. And the idea of the ILW procedure is utilizing the PDE repeatedly to write the normal derivatives of the variables \mathbf{W} in terms of the time derivatives and tangential derivatives of \mathbf{W} , both of which are obtained from boundary conditions. We should note that the boundary location varies with time, hence we will employ the material derivative rather than the Eulerian time derivative on the boundary. In the following subsections, we will concentrate on how to define the values of $W_{i,j,k}$ at the ghost points.

3.1 Boundary treatment for Euler equations

In the ILW method, we rewrite three-dimensional Euler equations in the form of primitive variables

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x + \mathbf{B}(\mathbf{U})\mathbf{U}_y + \mathbf{C}(\mathbf{U})\mathbf{U}_z = \mathbf{0}, \tag{16}$$

where

$$\mathbf{U} = (\rho, u, v, w, p)^T,$$

$\mathbf{A}(\mathbf{U})$, $\mathbf{B}(\mathbf{U})$ and $\mathbf{C}(\mathbf{U})$ are Jacobian matrixes, given as

$$\mathbf{A}(\mathbf{U}) = \begin{pmatrix} u & \rho & 0 & 0 & 0 \\ 0 & u & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & u & 0 & 0 \\ 0 & 0 & 0 & u & 0 \\ 0 & \rho c^2 & 0 & 0 & u \end{pmatrix}, \mathbf{B}(\mathbf{U}) = \begin{pmatrix} v & 0 & \rho & 0 & 0 \\ 0 & v & 0 & 0 & 0 \\ 0 & 0 & v & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & v & 0 \\ 0 & 0 & \rho c^2 & 0 & v \end{pmatrix}, \mathbf{C}(\mathbf{U}) = \begin{pmatrix} w & 0 & 0 & \rho & 0 \\ 0 & w & 0 & 0 & 0 \\ 0 & 0 & w & 0 & 0 \\ 0 & 0 & 0 & w & \frac{1}{\rho} \\ 0 & 0 & 0 & \rho c^2 & w \end{pmatrix},$$

and $c = \sqrt{\gamma p / \rho}$.

For any ghost point $P_{i,j,k} = (x_i, y_j, z_k)$, we first find a point $P_a = (x_a, y_a, z_a) \in \Gamma(t)$ so that its normal vector \mathbf{n} outside the rigid body would pass through $P_{i,j,k}$:

$$\begin{cases} x_a = x_c + r \sin \varphi \cos \theta, \\ y_a = y_c + r \sin \varphi \sin \theta, \\ z_a = z_c + r \cos \varphi. \end{cases} \tag{17}$$

The normal vector (from fluid to rigid body) of P_a is

$$\mathbf{n} = (-\sin \varphi \cos \theta, -\sin \varphi \sin \theta, -\cos \varphi).$$

Then, we rotate the coordinate axis locally at P_a such that the \hat{x} -axis of the new coordinate system is in the same direction with \mathbf{n} . We want to remark the fact that the rotation is not unique due to \hat{y} - and \hat{z} -axis can be any orthogonal vectors in the tangent plane with respect to the rigid body at P_a , but it will not effect the results via ILW procedure. Particularly, in Appendix B, we prove the following theorem:

Theorem 1 *The ILW boundary procedure for the Euler equation is independent of the choice of the rotation, i.e, we can get the same boundary scheme even with different choices of rotation.*

Therefore, we can get the same formulation even with different choices of tangent vectors. Here, we give one of these rotation methods and apply it to our later computation. The new local coordinate system $(\hat{x}, \hat{y}, \hat{z})$ and the coordinate transformation matrix \mathbf{T} are given as

$$\begin{pmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{pmatrix} = \mathbf{T} \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \text{and} \quad \mathbf{T} = \begin{pmatrix} -\cos \theta \cdot \sin \varphi & -\sin \theta \cdot \sin \varphi & -\cos \varphi \\ \sin \theta & -\cos \theta & 0 \\ -\cos \theta \cdot \cos \varphi & -\cos \varphi \cdot \sin \theta & \sin \varphi \end{pmatrix}. \quad (18)$$

Consequently, we can prove that the Euler Eq. (16) has the same form in the new coordinate system as those in the original coordinate system:

$$\hat{\mathbf{U}}_t + \mathbf{A}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{x}} + \mathbf{B}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{y}} + \mathbf{C}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{z}} = \mathbf{0}, \quad (19)$$

with

$$\hat{\mathbf{U}} = (\rho, \hat{u}, \hat{v}, \hat{w}, p)^T, \quad \text{and} \quad \begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{w} \end{pmatrix} = \mathbf{T} \begin{pmatrix} u \\ v \\ w \end{pmatrix}. \quad (20)$$

Next, we can approximate the values $\hat{\mathbf{U}}_{i,j,k}$ at point $P_{i,j,k}$ by a Taylor expansion at point P_a . For instance, a third order approximation is given as

$$\hat{\mathbf{U}}_{ijk} = \hat{\mathbf{U}}_a + (\mathbf{r} \cdot \mathbf{n})\hat{\mathbf{U}}_a^{(1)} + \frac{1}{2}(\mathbf{r} \cdot \mathbf{n})^2 \hat{\mathbf{U}}_a^{(2)}, \quad (21)$$

where \mathbf{r} is a vector going from P_a to $P_{i,j,k}$, $\hat{\mathbf{U}}_a$ (or $\hat{\mathbf{U}}_a^{(0)}$) is the point value at P_a , and $\hat{\mathbf{U}}_a^{(1)}$ and $\hat{\mathbf{U}}_a^{(2)}$ are the normal derivatives $\hat{\mathbf{U}}_{\hat{x}}|_{P_a}$ and $\hat{\mathbf{U}}_{\hat{x}\hat{x}}|_{P_a}$, respectively. It can be extended to arbitrary high order accuracy easily by keeping more terms in the Taylor expansion. For the current third-order scheme, we just need to get the values $\hat{\mathbf{U}}_a^{(m)}$, $m = 0, 1, 2$.

Before that, a local characteristic decomposition should be done at P_a , with the eigen-decomposition

$$\mathbf{A}(\hat{\mathbf{U}}_a) = \mathbf{L}_a^{-1} \Lambda_a \mathbf{L}_a,$$

where, Λ is the diagonal matrix consisting of the eigenvalues of $\mathbf{A}(\hat{\mathbf{U}})$, and \mathbf{L} forms from the corresponding left eigenvectors $\{\mathbf{l}_1, \dots, \mathbf{l}_5\}$:

$$\mathbf{L}_a = (\mathbf{l}_1, \dots, \mathbf{l}_5)^T = \begin{pmatrix} l_{1,1} & l_{1,2} & \cdots & l_{1,5} \\ l_{2,1} & l_{2,2} & \cdots & l_{2,5} \\ \vdots & \vdots & \ddots & \vdots \\ l_{5,1} & l_{5,2} & \cdots & l_{5,5} \end{pmatrix}.$$

Notice that the point value $\hat{\mathbf{U}}_a$ is unknown, we will use the WENO extrapolation to obtain the approximation $\hat{\mathbf{U}}_a^{ext}$ (see Appendix C) and then substitute it into the formulations above, obtaining $\mathbf{L}_a = \mathbf{L}(\hat{\mathbf{U}}_a^{ext})$. After that, convert the original variables $\hat{\mathbf{U}}$ to the characteristic fields $\mathbf{V} = \mathbf{L}_a \hat{\mathbf{U}}$. Since the eigenvalues of $\mathbf{A}(\hat{\mathbf{U}})$ are

$$\hat{u} - c, \hat{u}, \hat{u}, \hat{u}, \hat{u} + c,$$

the moving speeds of five characteristic lines relative to the boundary normal direction are $-c, 0, 0, 0, c$, respectively.

3.1.1 Reformulation of the material derivative in 3D

In the ILW procedure, we convert normal spatial derivatives to tangential and time derivatives of the given boundary condition using the PDE. Specifically, for moving boundaries, we would employ the material derivatives of the boundary conditions. Details will be shown in Section 3.1.2. For any function $\tilde{g} \in C^1(\mathbb{R}^3 \times \mathbb{R}^+)$, the material derivative is given as

$$\frac{D\tilde{g}}{Dt} = \frac{\partial \tilde{g}}{\partial t} + u \frac{\partial \tilde{g}}{\partial x} + v \frac{\partial \tilde{g}}{\partial y} + w \frac{\partial \tilde{g}}{\partial z}, \tag{22}$$

where, (u, v, w) is the flow velocity. Unfortunately, the definition above can not be used for the functions defined on boundary only, such as the normal vector, since the spatial derivatives may not exist. Therefore, for the functions limited on the surface in three-dimensional space, we will give a new definition of material derivative, which would play a key role in the ILW method in the following subsection.

Suppose that the three-dimensional moving boundary $\Gamma(t)$ is given as

$$\mathbf{X}_b = (x(\tau_1, \tau_2, t), y(\tau_1, \tau_2, t), z(\tau_1, \tau_2, t)),$$

where (τ_1, τ_2) are parameters in $[\tau_{10}, \tau_{11}] \times [\tau_{20}, \tau_{21}]$. For example, the surface of a unit sphere in 3D can be described in spherical coordinates:

$$x = \sin(\tau_1) \cos(\tau_2), \quad y = \sin(\tau_1) \sin(\tau_2), \quad z = \cos(\tau_2),$$

where, (τ_1, τ_2) are parameters in $[0, 2\pi] \times [-\pi/2, \pi/2]$.

Assume that, the parametric equation $\Gamma(t)$ is regular, i.e., $\frac{\partial \mathbf{X}_b}{\partial \tau_1}$ and $\frac{\partial \mathbf{X}_b}{\partial \tau_2}$ are linearly independent. Thus, $\frac{\partial \mathbf{X}_b}{\partial \tau_1}, \frac{\partial \mathbf{X}_b}{\partial \tau_2}$ and \mathbf{n} form a basis of the three-dimensional space. Then, the velocities of the fluid and the boundary can be written as follows:

$$\begin{aligned} \mathbf{u} &= u_n \mathbf{n} + u_{\tau_1} \frac{\partial \mathbf{X}_b}{\partial \tau_1} + u_{\tau_2} \frac{\partial \mathbf{X}_b}{\partial \tau_2}, \\ \mathbf{V}_b &= V_{b,n} \mathbf{n} + V_{b,\tau_1} \frac{\partial \mathbf{X}_b}{\partial \tau_1} + V_{b,\tau_2} \frac{\partial \mathbf{X}_b}{\partial \tau_2}. \end{aligned} \tag{23}$$

Definition 1 Suppose there is a function defined on $\Gamma(t)$: $g(\tau_1, \tau_2, t) \in C^1([\tau_{10}, \tau_{11}] \times [\tau_{20}, \tau_{21}] \times \mathbb{R}^+)$. The material derivative is defined as:

$$L(g) = \frac{\partial g}{\partial t} + (u_{\tau_1} - V_{b,\tau_1}) \frac{\partial g}{\partial \tau_1} + (u_{\tau_2} - V_{b,\tau_2}) \frac{\partial g}{\partial \tau_2}. \tag{24}$$

Moreover, it has the following property, which indicates the equivalence between our new definition (24) and the traditional definition of material derivative on no-penetration boundary.

Theorem 2 Suppose $\Gamma(t) = (x(\tau_1, \tau_2, t), y(\tau_1, \tau_2, t), z(\tau_1, \tau_2, t))$ is a smooth surface in three dimensions, and the no-penetration boundary condition holds on $\Gamma(t)$, i.e. $\mathbf{u} \cdot \mathbf{n} = \mathbf{V}_b \cdot \mathbf{n}$. For any $\tilde{g}(x, y, z, t) \in C^1(\mathbb{R}^3 \times \mathbb{R}^+)$, we can define a function g on $\Gamma(t)$:

$$g(\tau_1, \tau_2, t) = \tilde{g}(x(\tau_1, \tau_2, t), y(\tau_1, \tau_2, t), z(\tau_1, \tau_2, t), t). \tag{25}$$

Then the following equation is true:

$$L(g) = \frac{D\tilde{g}}{Dt} = \frac{\partial \tilde{g}}{\partial t} + \nabla \tilde{g} \cdot \mathbf{u}, \tag{26}$$

where $\frac{D}{Dt}$ is the material derivative given in (22).

Proof According to the formula (23) and $\mathbf{u} \cdot \mathbf{n} = \mathbf{V}_b \cdot \mathbf{n}$, we know that $u_n = V_{b,n}$. Using the chain rule, we can obtain that

$$\begin{aligned} \frac{\partial g}{\partial t} &= \frac{\partial \tilde{g}}{\partial t} + \frac{\partial \tilde{g}}{\partial x} \frac{\partial x}{\partial t} + \frac{\partial \tilde{g}}{\partial y} \frac{\partial y}{\partial t} + \frac{\partial \tilde{g}}{\partial z} \frac{\partial z}{\partial t} = \frac{\partial \tilde{g}}{\partial t} + \nabla \tilde{g} \cdot \mathbf{V}_b, \\ \frac{\partial g}{\partial \tau_1} &= \frac{\partial \tilde{g}}{\partial x} \frac{\partial x}{\partial \tau_1} + \frac{\partial \tilde{g}}{\partial y} \frac{\partial y}{\partial \tau_1} + \frac{\partial \tilde{g}}{\partial z} \frac{\partial z}{\partial \tau_1} = \nabla \tilde{g} \cdot \frac{\partial \mathbf{X}_b}{\partial \tau_1}, \\ \frac{\partial g}{\partial \tau_2} &= \frac{\partial \tilde{g}}{\partial x} \frac{\partial x}{\partial \tau_2} + \frac{\partial \tilde{g}}{\partial y} \frac{\partial y}{\partial \tau_2} + \frac{\partial \tilde{g}}{\partial z} \frac{\partial z}{\partial \tau_2} = \nabla \tilde{g} \cdot \frac{\partial \mathbf{X}_b}{\partial \tau_2}. \end{aligned}$$

Therefore,

$$\begin{aligned} L(g) &= \frac{\partial g}{\partial t} + (u_{\tau_1} - V_{b,\tau_1}) \frac{\partial g}{\partial \tau_1} + (u_{\tau_2} - V_{b,\tau_2}) \frac{\partial g}{\partial \tau_2} \\ &= \frac{\partial \tilde{g}}{\partial t} + \nabla \tilde{g} \cdot \mathbf{V}_b + (u_{\tau_1} - V_{b,\tau_1}) \nabla \tilde{g} \cdot \frac{\partial \mathbf{X}_b}{\partial \tau_1} \\ &\quad + (u_{\tau_2} - V_{b,\tau_2}) \nabla \tilde{g} \cdot \frac{\partial \mathbf{X}_b}{\partial \tau_2} + (u_n - V_{b,n}) \nabla \tilde{g} \cdot \mathbf{n} \\ &= \frac{\partial \tilde{g}}{\partial t} + \nabla \tilde{g} \cdot \mathbf{u} = \frac{D\tilde{g}}{Dt}. \end{aligned}$$

This completes the proof.

In the next subsection, we will use the definition L (24) on the given boundary conditions in the ILW process. For easy notation, we write $L(g)$ as $\frac{Dg}{Dt}$ in the following subsections.

3.1.2 ILW method

With the boundary conditions and the material derivative given in Definition 1, we now proceed to construct $\hat{\mathbf{U}}_a^{(m)} = ((\hat{U}_1)_a^{(m)}, \dots, (\hat{U}_5)_a^{(m)})^T, m = 0, 1, 2$.

- At first, we want to find the value of $\hat{\mathbf{U}}_a^{(0)}$. The no-penetration condition

$\hat{\mathbf{u}} = \hat{\mathbf{u}} \cdot \mathbf{n} = \mathbf{V}_b \cdot \mathbf{n}$ can be rewritten as

$$(\hat{U}_2)_a^{(0)} = \mathbf{V}_b \cdot \mathbf{n}.$$

Do the extrapolation of those outgoing characteristic variables, and we have the following equations

$$\mathbf{l}_j \cdot \hat{\mathbf{U}}_a^{(0)} = (V_j)_a^{(0)}, \quad j = 2, \dots, 5,$$

where, $(V_j)_a^{(0)}$ are obtained by the WENO extrapolation. Hence, those equations tell us a system as

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ l_{2,1} & l_{2,2} & l_{2,3} & l_{2,4} & l_{2,5} \\ l_{3,1} & l_{3,2} & l_{3,3} & l_{3,4} & l_{3,5} \\ l_{4,1} & l_{4,2} & l_{4,3} & l_{4,4} & l_{4,5} \\ l_{5,1} & l_{5,2} & l_{5,3} & l_{5,4} & l_{5,5} \end{pmatrix} \begin{pmatrix} (\hat{U}_1)_a^{(0)} \\ (\hat{U}_2)_a^{(0)} \\ (\hat{U}_3)_a^{(0)} \\ (\hat{U}_4)_a^{(0)} \\ (\hat{U}_5)_a^{(0)} \end{pmatrix} = \begin{pmatrix} \mathbf{V}_b \cdot \mathbf{n} \\ (V_2)_a^{(0)} \\ (V_3)_a^{(0)} \\ (V_4)_a^{(0)} \\ (V_5)_a^{(0)} \end{pmatrix}. \tag{27}$$

Thus, $\hat{\mathbf{U}}_a^{(0)}$ can be obtained by solving the above system.

- Then, we want to find the derivative of $\hat{\mathbf{U}}_a^{(1)}$. From the second equation of Euler equations with primitive variables we can get

$$p_{\hat{x}} = -\rho \frac{D\hat{u}}{Dt} = \rho \left[\hat{\mathbf{u}} \cdot \frac{D\hat{\mathbf{n}}}{Dt} - \frac{D}{Dt}(\mathbf{V}_b \cdot \mathbf{n}) \right],$$

where

$$\begin{aligned} \frac{D\mathbf{n}}{Dt} &= \frac{\mathbf{u} - \mathbf{V}_b}{r}, & \frac{D\hat{\mathbf{n}}}{Dt} &= \mathbf{T} \frac{D\mathbf{n}}{Dt}, \\ \frac{D}{Dt}(\mathbf{V}_b(t) \cdot \mathbf{n}) &= \frac{d\mathbf{V}_b}{dt} \cdot \mathbf{n} + \mathbf{V}_b \cdot \frac{D\mathbf{n}}{Dt}. \end{aligned}$$

So we get $\hat{\mathbf{U}}_a^{(1)}$ by solving the following system:

$$\begin{cases} (\hat{U}_5)_a^{(1)} = \rho \left(\hat{\mathbf{u}} \cdot \frac{D\hat{\mathbf{n}}}{Dt} - \frac{d}{dt}(\mathbf{V}_b \cdot \mathbf{n}) \right), \\ \mathbf{l}_j \cdot \hat{\mathbf{U}}_a^{(1)} = (V_j)_a^{(1)}, \quad j = 2, \dots, 5, \end{cases} \tag{28}$$

Again, $(V_j)_a^{(1)}$ are obtained by the WENO extrapolation.

- Finally, for the second derivative $\hat{\mathbf{U}}_a^{(2)}$, we will employ the idea of simplified ILW (SILW) procedure proposed in [20], and the second derivative can be obtained by WENO extrapolation.

Once $\hat{\mathbf{U}}_a^{(m)}$, $m = 0, 1, 2$ are obtained, we can use (21) to get $\hat{\mathbf{U}}_{i,j,k}$, and further $\mathbf{U}_{i,j,k}$ and $\mathbf{W}_{i,j,k}$.

3.1.3 Coupling with RK time discretization

Notice that the numerical method we have described previously is for time level t_n only. When coupling with the third order RK scheme (14), we need to match the time levels when constructing values of ghost points in the two intermediate stages $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$. In particular, the velocity \hat{u} at the boundary at each RK stage is necessary as the boundary condition, and we need to update \hat{u} with the following formulations instead of the RK scheme (14) to maintain the third order in time:

$$\begin{aligned} \hat{u}^{(1)} &= \hat{u}^n + \Delta t \frac{\partial \hat{u}}{\partial t} \Big|_{\mathbf{x}=\mathbf{X}_b(t_n), t=t_n}, \\ \hat{u}^{(2)} &= \hat{u}^n + \frac{\Delta t}{2} \frac{\partial \hat{u}}{\partial t} \Big|_{\mathbf{x}=\mathbf{X}_b(t_n), t=t_n} + \frac{\Delta t^2}{4} \frac{\partial^2 \hat{u}}{\partial t^2} \Big|_{\mathbf{x}=\mathbf{X}_b(t_n), t=t_n}. \end{aligned} \tag{29}$$

As in [18], we can use a standard Lax-Wendroff procedure to get the time derivatives as we want. More specifically, employing the second equation of (16) repeatedly, we can get the time derivatives $\frac{\partial \hat{u}}{\partial t}$ and $\frac{\partial^2 \hat{u}}{\partial t^2}$ in terms of spatial derivatives. After finishing an inverse Lax-Wendroff procedure at the time level t_n , all the first and second order spatial derivatives values at $\mathbf{x} = \mathbf{X}_b(t_n)$ are given. Hence, we can get $\frac{\partial \hat{u}}{\partial t}|_{\mathbf{x}=\mathbf{X}_b(t_n), t=t_n}$ and $\frac{\partial^2 \hat{u}}{\partial t^2}|_{\mathbf{x}=\mathbf{X}_b(t_n), t=t_n}$ with these spatial derivatives. More details can be found in [18].

3.2 Boundary treatments for NS equation

Similar to the case in Section 3.1, to define the value at a ghost point $P_{i,j,k} = (x_i, y_j, z_k)$, we set up a local coordinate system at P_a by (18) at first. Then, consider the variables and system in the local coordinate system. Again, the rotation is not unique but will not affect the results of boundary scheme. The proof is very similar to the Euler equation in Appendix B. We will not repeat it here.

Due to the fact that the Dirichlet boundary conditions are given for velocity $\hat{\mathbf{u}}$ and temperature T , we rewrite the Eq. (8) with respect to the new variables

$$\hat{\mathbf{U}} = (\rho, \hat{u}, \hat{v}, \hat{w}, T)^T,$$

and then,

$$\hat{\mathbf{U}}_t + \mathbf{D}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{x}} = \mathbf{E}(\hat{\mathbf{U}})\hat{\mathbf{U}}_{\hat{x}\hat{x}} + \mathbf{Res}, \tag{30}$$

The matrixes and source term are given as

$$\mathbf{D}(\hat{\mathbf{U}}) = \begin{pmatrix} \hat{u} & \rho & 0 & 0 & 0 \\ \frac{T}{\rho} & \hat{u} & 0 & 0 & 1 \\ 0 & 0 & \hat{u} & 0 & 0 \\ 0 & 0 & 0 & \hat{u} & 0 \\ 0 & (\gamma - 1)T & 0 & 0 & \hat{u} \end{pmatrix}, \quad \mathbf{E}(\hat{\mathbf{U}}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{4}{3Re \cdot \rho} & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{Re \cdot \rho} & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{Re \cdot \rho} & 0 \\ 0 & 0 & 0 & 0 & \frac{\gamma}{Pr \cdot Re \cdot \rho} \end{pmatrix}, \tag{31}$$

and

$$\mathbf{Res} = \begin{pmatrix} Res_1 \\ Res_2 \\ Res_3 \\ Res_4 \\ Res_5 \end{pmatrix} = \begin{pmatrix} -\rho(\hat{v}_{\hat{y}} + \hat{w}_{\hat{z}}) - \hat{v}\rho_{\hat{y}} - \hat{w}\rho_{\hat{z}} \\ \frac{1}{Re \cdot \rho}(\hat{u}_{\hat{y}\hat{y}} + \hat{u}_{\hat{z}\hat{z}} + \frac{1}{3}\hat{v}_{\hat{x}\hat{y}} + \frac{1}{3}\hat{w}_{\hat{x}\hat{z}}) - \hat{v}\hat{u}_{\hat{y}} - \hat{w}\hat{u}_{\hat{z}} \\ \frac{1}{Re \cdot \rho}(\frac{4}{3}\hat{v}_{\hat{y}\hat{y}} + \hat{v}_{\hat{z}\hat{z}} + \frac{1}{3}\hat{u}_{\hat{x}\hat{y}} + \frac{1}{3}\hat{w}_{\hat{y}\hat{z}}) - \frac{1}{\rho}P_{\hat{y}} - \hat{v}\hat{v}_{\hat{y}} - \hat{w}\hat{v}_{\hat{z}} \\ \frac{1}{Re \cdot \rho}(\hat{w}_{\hat{y}\hat{y}} + \frac{4}{3}\hat{w}_{\hat{z}\hat{z}} + \frac{1}{3}\hat{u}_{\hat{x}\hat{z}} + \frac{1}{3}\hat{v}_{\hat{y}\hat{z}}) - \frac{1}{\rho}P_{\hat{z}} - \hat{v}\hat{w}_{\hat{y}} - \hat{w}\hat{w}_{\hat{z}} \\ \frac{\gamma}{Re \cdot \rho \cdot Pr}(T_{\hat{y}\hat{y}} + T_{\hat{z}\hat{z}}) + NLT \end{pmatrix}, \tag{32}$$

with

$$NLT = \frac{\gamma - 1}{Re \cdot \rho} \left(-\frac{2}{3}(\hat{u}_{\hat{x}} + \hat{v}_{\hat{y}} + \hat{w}_{\hat{z}})^2 + 2(\hat{u}_{\hat{x}}^2 + \hat{v}_{\hat{y}}^2 + \hat{w}_{\hat{z}}^2) + 2(\hat{v}_{\hat{x}} + \hat{u}_{\hat{y}})^2 + 2(\hat{w}_{\hat{x}} + \hat{u}_{\hat{z}})^2 + 2(\hat{v}_{\hat{z}} + \hat{w}_{\hat{y}})^2 \right). \tag{33}$$

Again, the values $\hat{\mathbf{U}}_{i,j,k}$ at point $P_{i,j,k}$ will be approximated by a Taylor expansion in \hat{x} -direction at point P_a . Once obtaining the point value $\hat{\mathbf{U}}_a$, and normal derivatives $\hat{\mathbf{U}}_a^{(1)}$ and $\hat{\mathbf{U}}_a^{(2)}$, we can get the third order approximation of $\hat{\mathbf{U}}_{i,j,k}$ via (21).

Note that in the boundary treatment for the Euler equations in Section 3.1, the boundary values $\hat{\mathbf{U}}_a^{(m)}$ are obtained via either the ILW method based on the boundary conditions or the extrapolation of the outgoing variables. However, for the compressible NS equation, totally different boundary treatments are needed for the diffusion-dominated and the convection-dominated regimes. Hence, [28] designed the numerical methods based on a

Careful combination of the boundary treatments for the two regimes and obtained a stable and accurate boundary condition for general convection-diffusion equations with either still or moving boundaries. Here, we extend the numerical method to our 3D model with the given isothermal no-slip wall boundary condition.

3.2.1 The material derivative in 3D

For the NS equation with moving boundary, the material derivative is also necessary in the ILW method.

$$\frac{D\hat{\mathbf{U}}}{Dt} = \hat{\mathbf{U}}_t + \hat{u} \hat{\mathbf{U}}_{\hat{x}} + \hat{v} \hat{\mathbf{U}}_{\hat{y}} + \hat{w} \hat{\mathbf{U}}_{\hat{z}}.$$

For the no-slip boundary, the material derivatives for the boundary and fluid are the same,

$$\frac{D\hat{\mathbf{U}}}{Dt} = \frac{D\hat{\mathbf{U}}_b}{Dt}.$$

Moreover, since T_b is constant, hence $DT_b/Dt = 0$. And $D\hat{\mathbf{u}}/Dt = d\mathbf{V}_b/dt$ equal to the acceleration of the rigid body. As a consequence,

$$\begin{cases} \frac{D\rho}{Dt} + \rho(\hat{u}_{\hat{x}} + \hat{v}_{\hat{y}} + \hat{w}_{\hat{z}}) = 0 \\ \frac{D\hat{u}}{Dt} + \frac{1}{\rho} p_{\hat{x}} = \frac{1}{Re\rho} \left(\frac{4}{3} \hat{u}_{\hat{x}\hat{x}} + \hat{u}_{\hat{y}\hat{y}} + \hat{u}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{v}_{\hat{x}\hat{y}} + \frac{1}{3} \hat{w}_{\hat{x}\hat{z}} \right) \\ \frac{D\hat{v}}{Dt} + \frac{1}{\rho} p_{\hat{y}} = \frac{1}{Re\rho} \left(\hat{v}_{\hat{x}\hat{x}} + \frac{4}{3} \hat{v}_{\hat{y}\hat{y}} + \hat{v}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{u}_{\hat{x}\hat{y}} + \frac{1}{3} \hat{w}_{\hat{y}\hat{z}} \right) \\ \frac{D\hat{w}}{Dt} + \frac{1}{\rho} p_{\hat{z}} = \frac{1}{Re\rho} \left(\hat{w}_{\hat{x}\hat{x}} + \hat{w}_{\hat{y}\hat{y}} + \frac{4}{3} \hat{w}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{u}_{\hat{x}\hat{z}} + \frac{1}{3} \hat{v}_{\hat{y}\hat{z}} \right) \\ \frac{DT}{Dt} + (\gamma - 1)T(\hat{u}_{\hat{x}} + \hat{v}_{\hat{y}} + \hat{w}_{\hat{z}}) = \frac{\gamma}{\rho Re Pr} (T_{\hat{x}\hat{x}} + T_{\hat{y}\hat{y}} + T_{\hat{z}\hat{z}}) + NLT \end{cases} \tag{34}$$

3.2.2 ILW method

In particular, the derivatives $\hat{\mathbf{U}}_a^{(m)}$, $m = 1, 2$, are a combination of those derived from the ILW procedure and the extrapolation procedure from interior points. For easy notation, we use the subscript “ilw” and “ext” to denote derivatives $\hat{\mathbf{U}}_a^{(m)}$ obtained through the ILW procedure $\hat{\mathbf{U}}_{a,ilw}^{(m)}$ and the extrapolation $\hat{\mathbf{U}}_{a,ext}^{(m)}$, respectively.

Here, the matrix $\mathbf{D}(\hat{\mathbf{U}})$ is also diagonalizable,

$$\mathbf{D}(\hat{\mathbf{U}}) = \mathbf{L}^{-1}(\hat{\mathbf{U}})\Lambda(\hat{\mathbf{U}})\mathbf{L}(\hat{\mathbf{U}}),$$

with $\Lambda(\hat{\mathbf{U}}) = \text{diag}\{\hat{u} - c, \hat{u}, \hat{u}, \hat{u} + c\}$. Hence, considering the characteristic variables $\mathbf{V} = \mathbf{L}_a \hat{\mathbf{U}}$, the components V_2, \dots, V_5 are the outflow variables, and V_1 is the inflow variable. Again, $\hat{\mathbf{U}}_a^{ext}$ is used in the characteristic decomposition.

- Construct the point value $\hat{\mathbf{U}}_a^{(0)}$. Note that the boundary condition (9) gives the value of $\hat{\mathbf{u}}$ and T at the boundary $\Gamma(t)$ directly, so we can set

$$(\hat{U}_2)_a^{(0)} = \hat{u}_b, \quad (\hat{U}_3)_a^{(0)} = \hat{v}_b, \quad (\hat{U}_4)_a^{(0)} = \hat{w}_b, \quad (\hat{U}_5)_a^{(0)} = T_b. \tag{35}$$

Since V_5 is the outgoing variable, we have the relation that

$$\mathbf{I}_5 \cdot \hat{\mathbf{U}}_a^{(0)} = V_5 = \mathbf{I}_5 \cdot \hat{\mathbf{U}}_{a,ext}^{(0)}.$$

Therefore, we can obtain the value of $(\hat{U}_1)_a^{(0)}$ as follows,

$$(\hat{U}_1)_a^{(0)} = \frac{(\hat{U}_1)_{a,ext}^{(0)}}{(\hat{U}_5)_{a,ext}^{(0)}} \left(2 (\hat{U}_5)_{a,ext}^{(0)} - (\hat{U}_5)_a^{(0)} + \sqrt{\gamma \cdot (\hat{U}_5)_{a,ext}^{(0)}} ((\hat{U}_2)_{a,ext}^{(0)} - (\hat{U}_2)_a^{(0)}) \right). \tag{36}$$

- Construct the derivative value $\hat{U}_a^{(1)}$. Combining the definition $p = \rho T$ and material derivative $D\hat{u}/Dt$ in (34), we have that

$$(\hat{U}_5)_a^{(0)} (\hat{U}_1)_{a,ilw}^{(1)} + (\hat{U}_1)_a^{(0)} (\hat{U}_5)_{a,ilw}^{(1)} = -(\hat{U}_1)_a^{(0)} \frac{D\hat{u}_b}{Dt} + \frac{1}{Re} \left(\frac{4}{3} \hat{u}_{\hat{x}\hat{x}} + \hat{u}_{\hat{y}\hat{y}} + \hat{u}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{v}_{\hat{x}\hat{y}} + \frac{1}{3} \hat{w}_{\hat{x}\hat{z}} \right).$$

Given the outgoing performance of V_2, \dots, V_5 , we have the relation that

$$\mathbf{I}_j \cdot \hat{\mathbf{U}}_{a,ilw}^{(1)} = (V_j)_a^{(1)}, \quad j = 2, \dots, 5.$$

The second derivatives on the right hand side and $(V_j)_a^{(1)}$ are obtained via extrapolation. Thus, we can have $\hat{\mathbf{U}}_{a,ilw}^{(1)}$ by solving the system. Then,

$$\hat{\mathbf{U}}_a^{(1)} = \frac{c^2 h^2}{c^2 h^2 + 9\epsilon^2} \hat{\mathbf{U}}_{a,ilw}^{(1)} + \frac{9\epsilon^2}{c^2 h^2 + 9\epsilon^2} \hat{\mathbf{U}}_{a,ext}^{(1)}. \tag{37}$$

with $\epsilon = \max\{\frac{1}{Re\rho}, \frac{4}{3Re\rho}, \frac{\gamma}{PrRe\rho}\}$ and $h = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$.

- Construct the derivative value $\hat{\mathbf{U}}_a^{(2)}$. Based on the boundary condition (34), the second derivatives $(\hat{U}_2)^{(2)}, \dots, (\hat{U}_5)^{(2)}$ satisfy equations and we set them as "ilw":

$$\left\{ \begin{aligned} (\hat{U}_2)_{a,ilw}^{(2)} &= \frac{3Re\rho}{4} \left(\frac{D\hat{u}_b}{Dt} + \frac{1}{\rho} P_{\hat{x}} - \frac{1}{Re\rho} (\hat{u}_{\hat{y}\hat{y}} + \hat{u}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{v}_{\hat{x}\hat{y}} + \frac{1}{3} \hat{w}_{\hat{x}\hat{z}}) \right), \\ (\hat{U}_3)_{a,ilw}^{(2)} &= Re\rho \left(\frac{D\hat{v}_b}{Dt} + \frac{1}{\rho} P_{\hat{y}} - \frac{1}{Re\rho} \left(\frac{4}{3} \hat{v}_{\hat{y}\hat{y}} + \hat{v}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{u}_{\hat{x}\hat{y}} + \frac{1}{3} \hat{w}_{\hat{y}\hat{z}} \right) \right), \\ (\hat{U}_4)_{a,ilw}^{(2)} &= Re\rho \left(\frac{D\hat{w}_b}{Dt} + \frac{1}{\rho} P_{\hat{z}} - \frac{1}{Re\rho} (\hat{w}_{\hat{y}\hat{y}} + \frac{4}{3} \hat{w}_{\hat{z}\hat{z}} + \frac{1}{3} \hat{u}_{\hat{x}\hat{z}} + \frac{1}{3} \hat{v}_{\hat{y}\hat{z}}) \right), \\ (\hat{U}_5)_{a,ilw}^{(2)} &= \frac{Re\rho Pr}{\gamma} \left(\frac{DT_b}{Dt} (\gamma - 1) T (\hat{u}_{\hat{x}} + \hat{v}_{\hat{y}} + \hat{w}_{\hat{z}}) - \frac{\gamma}{Re\rho Pr} (T_{\hat{y}\hat{y}} + T_{\hat{z}\hat{z}}) - NLT \right). \end{aligned} \right.$$

There is no diffusion term in the first equation of (34), so we just take

$$(\hat{U}_1)_{a,ilw}^{(2)} = (\hat{U}_1)_{a,ext}^{(2)},$$

we can solve the system to get $\hat{\mathbf{U}}_{a,ilw}^{(2)}$. Finally, we have that

$$(\hat{U}_j)_a^{(2)} = \frac{c^2 h^2}{c^2 h^2 + 9\epsilon_j^2} (\hat{U}_j)_{a,ext}^{(2)} + \frac{9\epsilon_j^2}{c^2 h^2 + 9\epsilon_j^2} (\hat{U}_j)_{a,ilw}^{(2)}, \tag{38}$$

with $\{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5\} = \left\{ 0, \frac{4}{3Re\rho}, \frac{1}{Re\rho}, \frac{1}{Re\rho}, \frac{\gamma}{PrRe\rho} \right\}$.

3.2.3 Coupling with RK time discretization

When coupling the third order SSP RK time discretization, we need the value of $(\hat{U}_j)_t|_{P_a, t_n}$ and $(\hat{U}_j)_{tt}|_{P_a, t_n}, j = 2, \dots, 5$, to obtain high order accuracy in time via (29).

Fix at the boundary $(x, y, z) = P_a$ and $t = t_n$,

$$\frac{D\hat{U}_j}{Dt} = (\hat{U}_j)_t + \hat{u}_b (\hat{U}_j)_{\hat{x}} + \hat{v}_b (\hat{U}_j)_{\hat{y}} + \hat{w}_b (\hat{U}_j)_{\hat{z}},$$

we can take $(\hat{U}_j)_{\hat{x}} = (\hat{U}_j)_a^{(1)}$ which is obtained from the previous boundary treatment, and $(\hat{U}_j)_{\hat{y}}$ and $(\hat{U}_j)_{\hat{z}}$ can be approximated through extrapolation. Moreover, from the

previous section, we can see that for the compressible viscous fluid $\frac{D\hat{U}_j}{Dt}$ at boundary is given via boundary conditions. Hence, we can have $\frac{D\hat{U}_j}{Dt}$ directly. Thus, for $j = 2, \dots, 5$,

$$\left(\hat{U}_j\right)_t \Big|_{(x,y,z)=P_a,t=t_n} = \left(\frac{D\hat{U}_j}{Dt} - \hat{u}_b \left(\hat{U}_j\right)^{(1)} - \hat{v}_b \left(\hat{U}_j\right)_{\hat{y}} - \hat{w}_b \left(\hat{U}_j\right)_{\hat{z}}\right) \Big|_{(x,y,z)=P_a,t=t_n}. \quad (39)$$

Look at the second material derivative

$$\begin{aligned} \frac{D^2\hat{U}_j}{Dt^2} &= \frac{\partial\hat{u}_b}{\partial t} \left(\hat{U}_j\right)_{\hat{x}} + \frac{\partial\hat{v}_b}{\partial t} \left(\hat{U}_j\right)_{\hat{y}} + \frac{\partial\hat{w}_b}{\partial t} \left(\hat{U}_j\right)_{\hat{z}} + \hat{u}_b^2 \left(\hat{U}_j\right)_{\hat{x}\hat{x}} + \hat{v}_b^2 \left(\hat{U}_j\right)_{\hat{y}\hat{y}} + \hat{w}_b^2 \left(\hat{U}_j\right)_{\hat{z}\hat{z}} \\ &\quad + 2\hat{u}_b\hat{v}_b \left(\hat{U}_j\right)_{\hat{x}\hat{y}} + 2\hat{u}_b\hat{w}_b \left(\hat{U}_j\right)_{\hat{x}\hat{z}} + 2\hat{v}_b\hat{w}_b \left(\hat{U}_j\right)_{\hat{y}\hat{z}} + 2\hat{u}_b \left(\hat{U}_j\right)_{t\hat{x}} + 2\hat{v}_b \left(\hat{U}_j\right)_{t\hat{y}} \\ &\quad + 2\hat{w}_b \left(\hat{U}_j\right)_{t\hat{z}} + \left(\hat{U}_j\right)_{tt}. \end{aligned}$$

Here, $\left(\frac{\partial\hat{u}_b}{\partial t}, \frac{\partial\hat{v}_b}{\partial t}, \frac{\partial\hat{w}_b}{\partial t}\right)^T$ are the acceleration of the boundary. In particular, the spatial derivatives $\left(\hat{U}_j\right)_{\alpha}$ and $\left(\hat{U}_j\right)_{\alpha\beta}$ (α, β can be \hat{x}, \hat{y} or \hat{z}) at (P_a, t_n) are already obtained via the ILW process t_n . Hence, we only need to deal with $\left(\hat{U}_j\right)_{t\hat{x}}, \left(\hat{U}_j\right)_{t\hat{y}}$ and $\left(\hat{U}_j\right)_{t\hat{z}}$. Those values can be simulated through the standard Lax-Wendroff idea, i.e., turning the time derivative into spatial derivatives via PDE, and then approximating the spatial ones. However, the process would be very complex and the computational cost is large. Notice that the time derivative $\left(\hat{U}_j\right)_t \Big|_{i,j,k}$ at time t_n is already known. Hence, the mixed derivatives $\left(\hat{U}_j\right)_{t\hat{x}}, \left(\hat{U}_j\right)_{t\hat{y}}$ and $\left(\hat{U}_j\right)_{t\hat{z}}$ can be approximated by the spatial extrapolation.

4 Description of the moving rigid body

Under the impingement of the shock, the rigid body will move in the fluid. Here, both translation and rotation of the body should be taken into account. For each point $\mathbf{X}_b = (x_b, y_b, z_b) \in \Gamma(t)$, the velocity is given as

$$\mathbf{V}_b = \mathbf{V}_{tr} + \omega \times \mathbf{r},$$

where, \mathbf{V}_{tr} is the translational velocity, ω is the rotation velocity, and \mathbf{r} is the vector from the boundary point to the centroid coordinates $\mathbf{X}_c(t) = (x_c(t), y_c(t), z_c(t))$. Furthermore, \mathbf{V}_{tr} and ω satisfy the equation

$$\frac{d\mathbf{X}_b}{dt} = \mathbf{V}_b, \quad \frac{d\mathbf{V}_{tr}}{dt} = \mathbf{a}_{tr}, \quad \frac{d\omega}{dt} = \mathbf{a}_{\theta}, \quad (40)$$

with the translational acceleration \mathbf{a}_{tr} and rotational acceleration \mathbf{a}_{θ} determined by Newton's second law and rotational law respectively. In particular, for the inviscid fluid,

$$\begin{aligned} M\mathbf{a}_{tr} &= \oint_{\Gamma(t)} p \mathbf{n} dS, \\ I\mathbf{a}_{\theta} &= \oint_{\Gamma(t)} p \mathbf{r} \times \mathbf{n} dS. \end{aligned} \quad (41)$$

Here, M is the mass of the sphere, \mathbf{n} is the external normal vector of the surface (from the rigid body to the fluid), and I is the moment of inertia of the sphere. While, for the viscous fluid,

$$\begin{aligned} M\mathbf{a}_{tr} &= \oint_{\Gamma(t)} (p \mathbf{n} - \boldsymbol{\tau} \cdot \mathbf{n}) dS, \\ I\mathbf{a}_{\theta} &= \oint_{\Gamma(t)} \mathbf{r} \times (p \mathbf{n} - \boldsymbol{\tau} \cdot \mathbf{n}) dS, \end{aligned} \quad (42)$$

with τ being the matrix defined as $\tau = \frac{1}{Re} [\tau_{ij}]_{3 \times 3}$.

For the surface integrals in (41) and (42), these integrals can be expressed in the double integral form with spherical coordinates. More specifically, we divide the integration area into uniform grids in two directions, and then can use the two points composite Gaussian quadrature to integrate these double integrals. And the integrand can be obtained by extrapolation at each Gaussian point.

5 Numerical results

In this section, we will show a series of numerical examples. Time step is taken as:

$$\Delta t = \min \left(\frac{CFL}{\lambda_x/\Delta x + \lambda_y/\Delta y + \lambda_z/\Delta z}, \frac{\Delta x}{\Gamma \max(|V_{b,x}|)}, \frac{\Delta y}{\Gamma \max(|V_{b,y}|)}, \frac{\Delta z}{\Gamma \max(|V_{b,z}|)} \right), \tag{43}$$

for Euler equation, and

$$\Delta t = \min \left(\frac{CFL}{\lambda_x/\Delta x + \lambda_y/\Delta y + \lambda_z/\Delta z + \lambda_d(1/\Delta x^2 + 1/\Delta y^2 + 1/\Delta z^2)}, \frac{\Delta x}{\Gamma \max(|V_{b,x}|)}, \frac{\Delta y}{\Gamma \max(|V_{b,y}|)}, \frac{\Delta z}{\Gamma \max(|V_{b,z}|)} \right), \tag{44}$$

for NS equation, with $\lambda_x = \max_{ijk} |u_{i,j,k}| + c$, $\lambda_y = \max_{ijk} |v_{i,j,k}| + c$, $\lambda_z = \max_{ijk} |w_{i,j,k}| + c$, and $\lambda_d = \max(\frac{1}{Re\rho}, \frac{4}{3Re\rho}, \frac{\gamma}{PrRe\rho})$. We take $CFL = 0.3$ in our numerical simulation, and the constraint (15) is automatically satisfied.

5.1 An example of accuracy test

In this example, we construct an inviscid isentropic flow and measure the entropy errors to verify the accuracy of our algorithm. This model is obtained by stretching the 2D model (see Example 4.1 in [13]) in z -direction. But we do use the 3D algorithm introduced in this paper to solve this problem. Specifically, the 3D example is given as follows:

The governing equation is the Euler Eq. (6). The computational domain is taken as $[-4, 4] \times [-4, 4] \times [-0.5, 0.5]$. The boundaries in x - and y -direction are all solid walls. A moving rigid cylinder with radius $R = 1.0$ and mass $M = 1.0$ is initially placed in the domain paralleling to the z -axis, and the central axis is located at $x = y = 0$. It will move under the impingement of inviscid flow in the domain. The initial flow should be isentropic, smooth and consistent with the boundary condition such that the flow keeps isentropic as long as the solution stays smooth. Therefore, we set

$$\begin{aligned} u(x, y, z, 0) &= -\frac{1}{2} (\lambda_1(x, y)u_1(x, y) + \lambda_2(x, y)^2), \\ v(x, y, z, 0) &= \frac{1}{2} (\lambda_1(x, y)v_1(x, y) + \lambda_2(x, y)v_2(x, y)), \\ w(x, y, z, 0) &= 0, \\ p(x, y, z, 0) &= 1, \\ \rho(x, y, z, 0) &= 1, \end{aligned}$$

where

$$\lambda_1(x, y) = \frac{(4\sqrt{2} - 1)(\sqrt{x^2 + y^2} - 1)}{(\sqrt{16 + y^2} - 1)(\sqrt{16 + x^2} - 1)},$$

$$\lambda_2(x, y) = \frac{\sqrt{x^2 + y^2} (x^2 - 16) (y^2 - 16)}{\left(\frac{x^2}{x^2 + y^2} - 16\right) \left(\frac{y^2}{x^2 + y^2} - 16\right)},$$

and

$$u_1(x, y) = \sin\left(\frac{\pi}{4}x\right) \sin^2\left(\frac{\pi}{4}y\right),$$

$$v_1(x, y) = \sin^2\left(\frac{\pi}{4}x\right) \sin\left(\frac{\pi}{4}y\right),$$

$$v_2(x, y) = \frac{1}{16}(x^2 + y^2 - 1) \sin\left(\frac{\pi}{4}x\right),$$

and we set the initial velocity of the moving cylinder as $V_b = (-0.5, 0, 0)^T$.

As long as the solution stays smooth, it maintains the isentropic flow, i.e., $s(x, y, z, t) = 1$. In particular, solutions on any xy -section plane are the same as those in 2D. Hence, we use zero spatial derivative on the boundaries in z -direction to guarantee two dimensional property of the flow around the cylinder. We use the reflection boundary condition on the boundaries in x - and y -direction, and the moving boundary algorithm proposed in this paper is applied to deal with the cylinder boundary.

In Table 1, we give the entropy errors and the rate of convergence at $t = 0.5$ when the solution is smooth. It is observed that the scheme can achieve the designed third order accuracy.

5.2 Interaction between shock wave and a cylinder

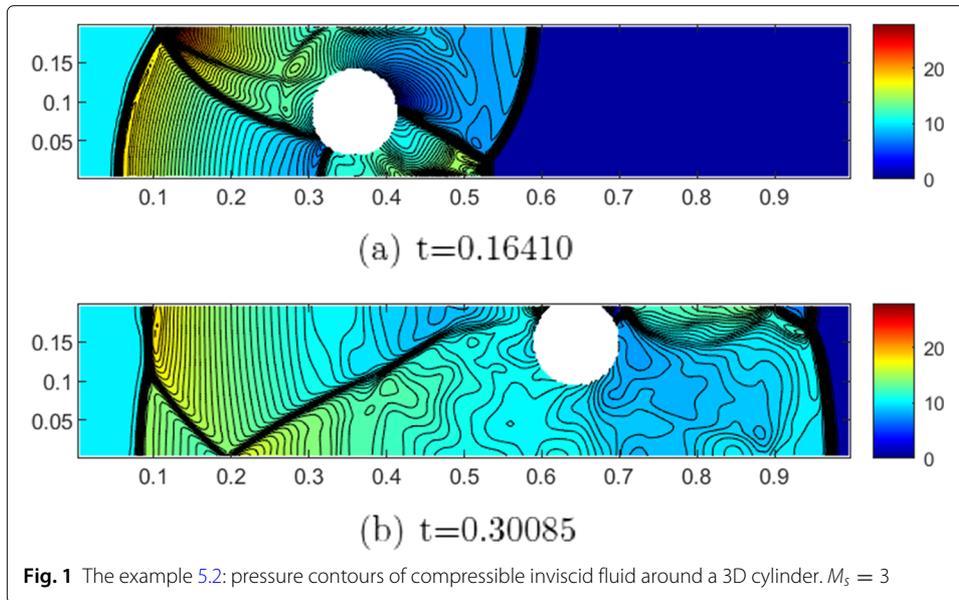
This example is the three dimensional version of Example 4.3 in [13] and Example 5 in [28]. We use it to verify the correctness of our algorithm. As in the two papers, the computation area is taken as $[x_l, x_r] \times [y_l, y_r] \times [z_l, z_r] = [0, 1] \times [0, 0.2] \times [-0.5, 0.5]$. At the initial time, a cylinder is parallel to y -direction with its center located at $(0.15, 0.05, 0)$. The radius and the density are given as $r = 0.05$ and $\sigma = 10.77$, respectively. A shock wave is located at $x = 0.08$. We test the example with Mach number $M_s = 3$. In z -direction, we use zero spatial derivative on the infinite boundaries to guarantee two dimensional property of the flow around the cylinder.

- **Compressible inviscid fluid.** We plot the pressure contour in Fig. 1 along the cross section of $z = 0$ at two typical time points.
- **Compressible viscous fluid.** We take $Re = 1000$, $Pr = 0.7$, and $T_b = \frac{5}{7}$. We plot the pressure contour in Fig. 2 along the cross section of $z = 0$ at two typical time points.

Our numerical results consist with those of two dimensional ones in [13, 28], indicating our method for three dimensional problems is correct.

Table 1 The example 5.1: entropy errors and convergence rates

h	L^1 error	Order	L^∞ error	Order
1/5	7.47E-002	–	1.14E-002	–
1/10	9.57E-003	2.97	2.21E-003	2.37
1/20	1.03E-003	3.22	2.73E-004	3.02
1/40	1.11E-004	3.21	3.14E-005	3.12

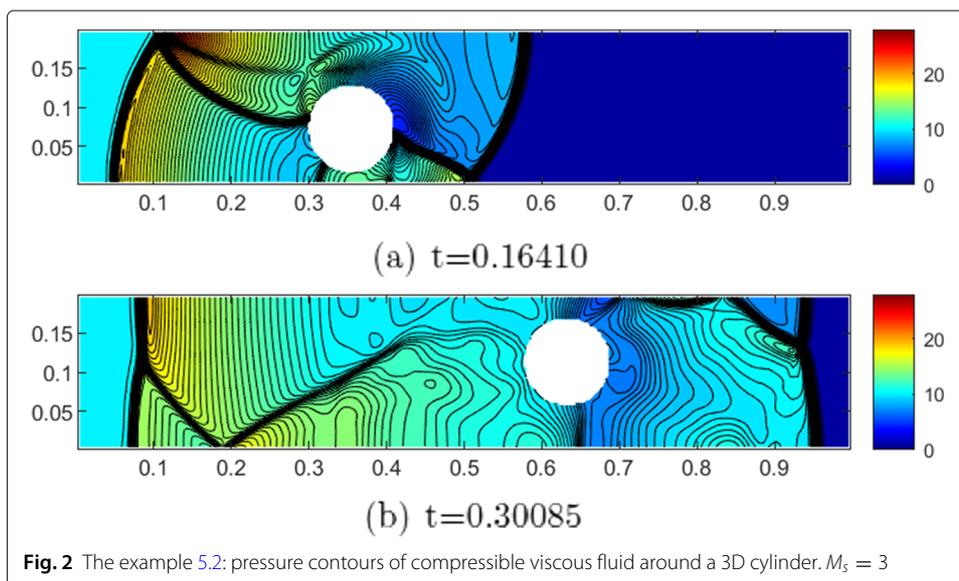


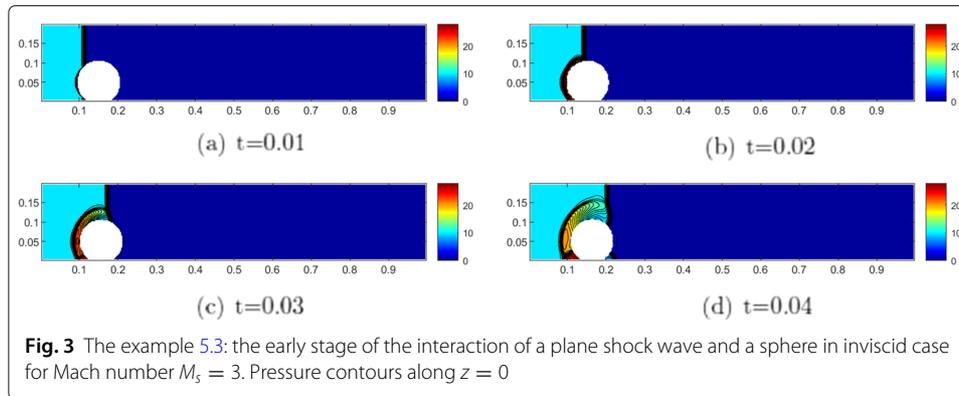
5.3 Interaction between shock wave and a sphere

The computational domain is taken as $[x_l, x_r] \times [y_l, y_r] \times [z_l, z_r] = [0, 1] \times [0, 0.2] \times [-0.5, 0.5]$. The initial position of the spherical center is $(0.15, 0.05, 0)$, with the radius $r = 0.05$ and the density $\sigma = 10.77$. And a shock wave is located at $x = 0.08$ initially. We test the example with $M_s = 3$ and $M_s = 6$.

In the early stage of computation, the sphere is impacted by the shock wave to produce a reflected shock wave, and the reflected shock wave hits the lower wall to generate a new reflected shock wave, causing uneven pressure distribution on the upper and lower sides of the sphere. The uneven pressure causes the sphere to move upward. Figures 3 and 4 show this procedure for inviscid fluid and viscous fluid respectively.

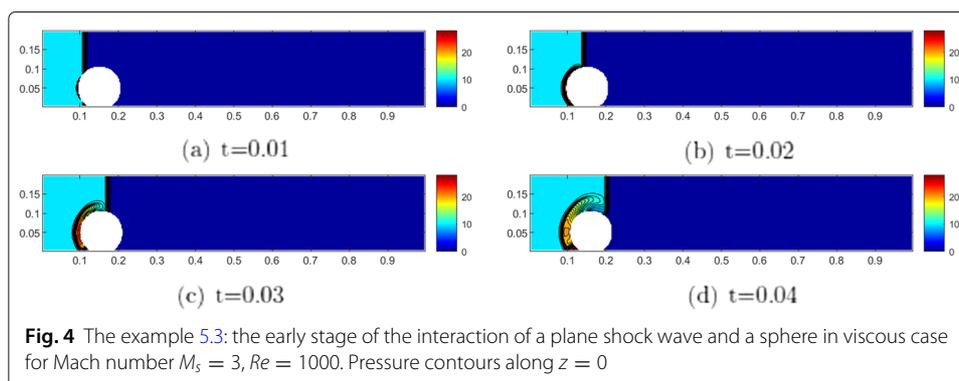
Later, the reflections of Mach shocks from the walls reach the sphere, generating more structures. Those structures will vary with the fluid.

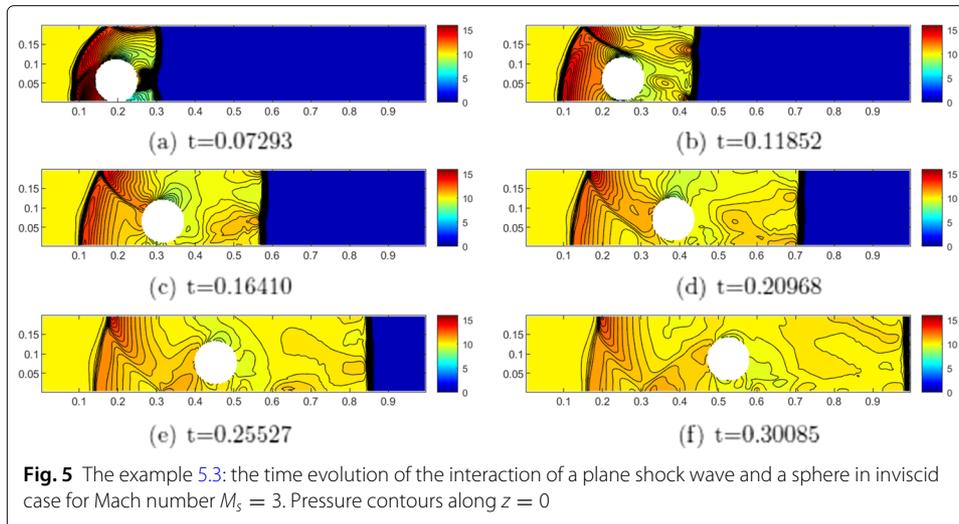




- Compressible inviscid fluid.** Figures 5, 6 and 7 contain the time evolution of the interaction of a plane shock wave and a sphere for Mach number $M_s = 3$ and $M_s = 6$ respectively. The pressure contours along the cross section of $z = 0$ are plotted. We can observe that there is a reflected shock wave when the incident shock wave interacts with the sphere. There is a second reflected shock wave when the reflected shock wave interacts with the upper wall. The second reflected shock wave hits the sphere again and forms a secondary interaction. The rigid body would be under greater stress in fluid with a larger Mach number. Consequently, the body would move faster. These numerical results show that our numerical method is robust in the computation for the supersonic flow with high Mach number.
- Compressible viscous fluid.** We take $Re = 500$ and 10^7 , $Pr = 0.7$, and $T_b = \frac{5}{7}$. The time evolution of the interaction of a plane shock wave and a sphere for Mach number $M_s = 3$ and $M_s = 6$ is plotted in Figs. 8, 9, 10, 11 and 12. Again, the system of shock waves is observed clearly, and results are significantly different with different Mach numbers. However, there is no discernible difference when changing Reynolds number.

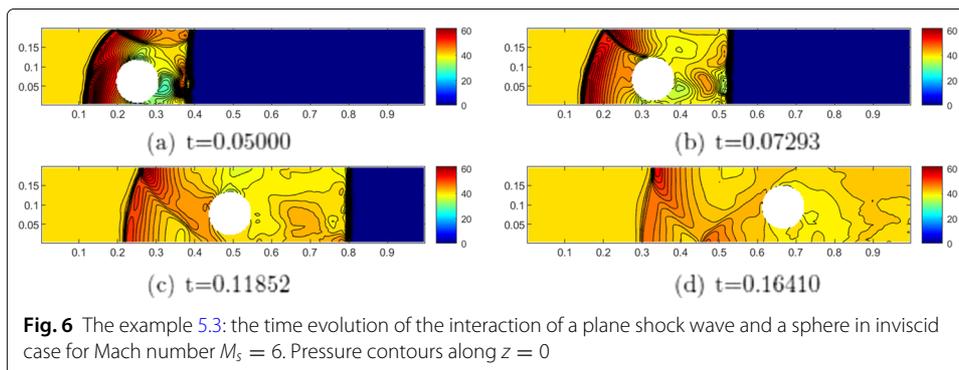
In addition, 3D pictures of the iso-surface of vorticity and the contour of pressure of the inviscid fluid and viscous fluid are given in Figs. 7 and 12, respectively. The numerical results demonstrate that our methods are stable and efficient for general cases.

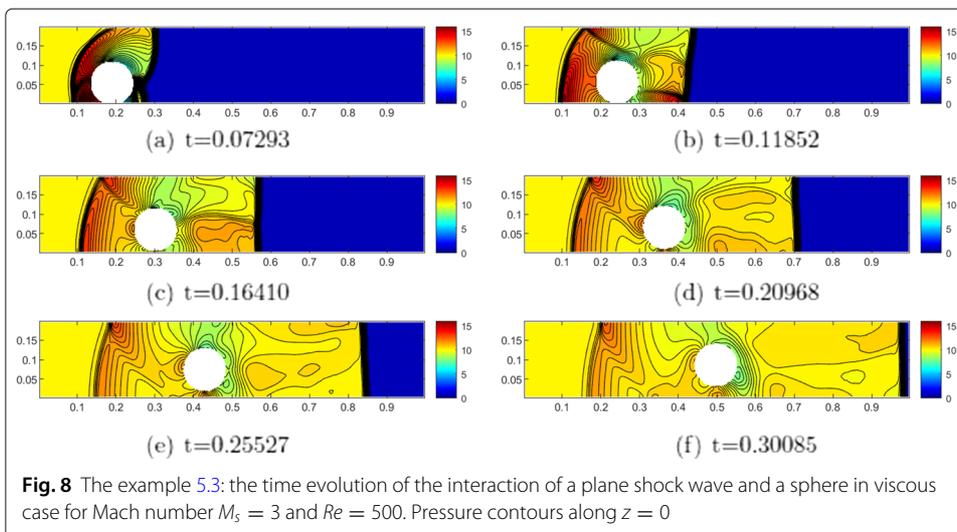
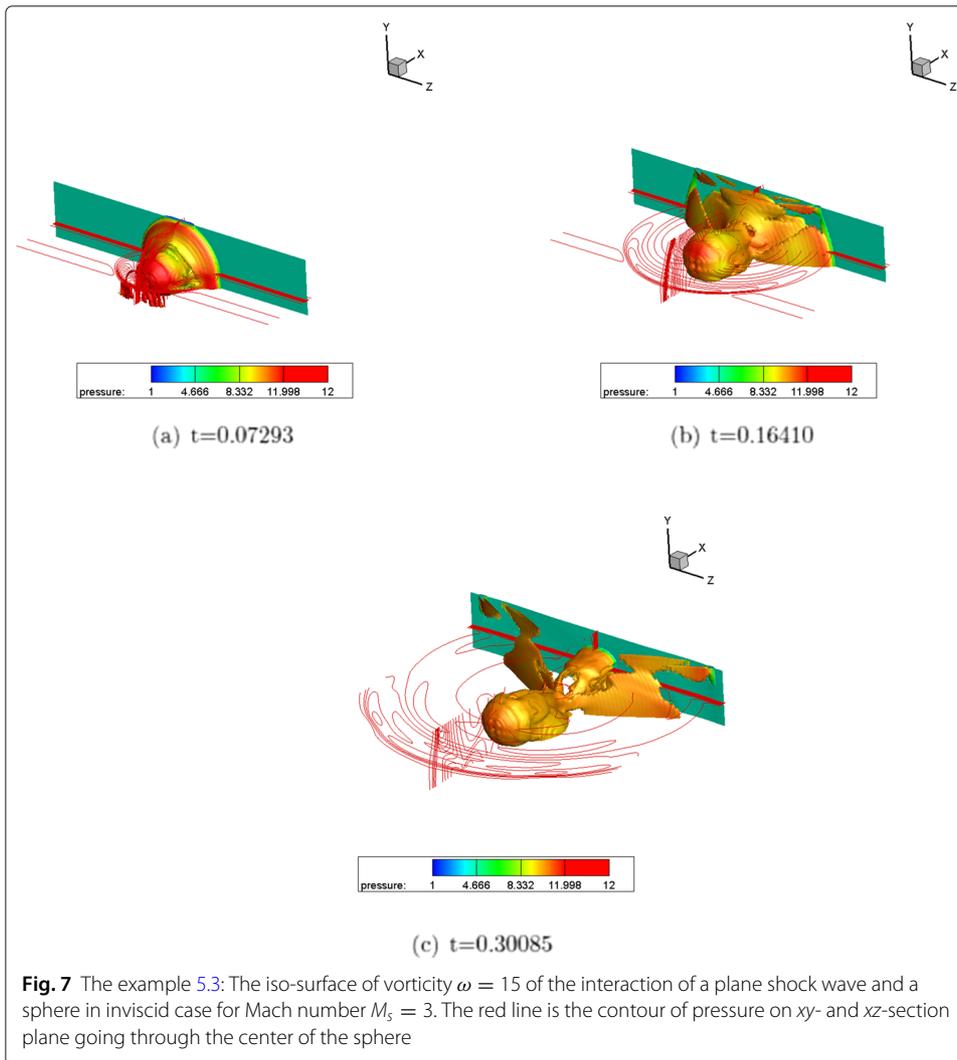


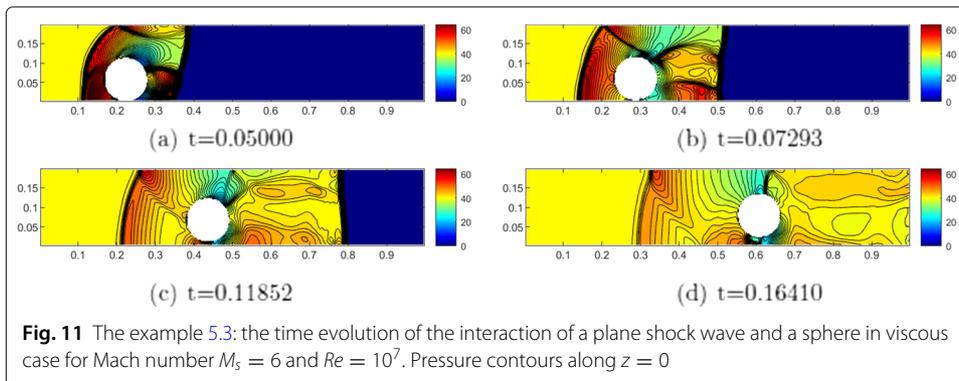
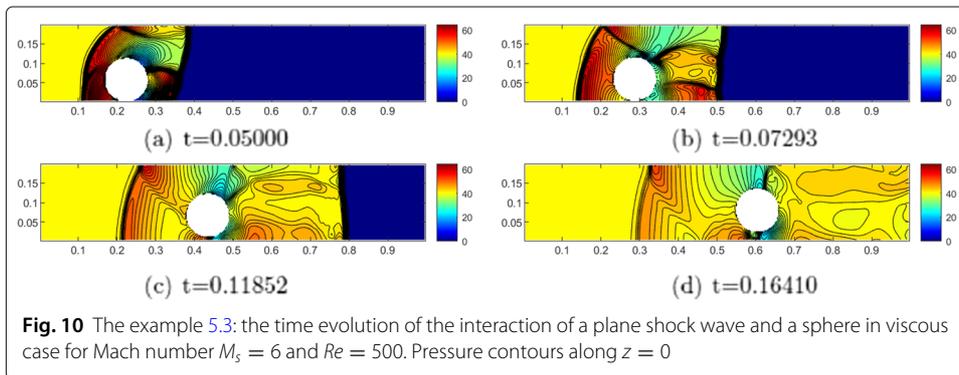
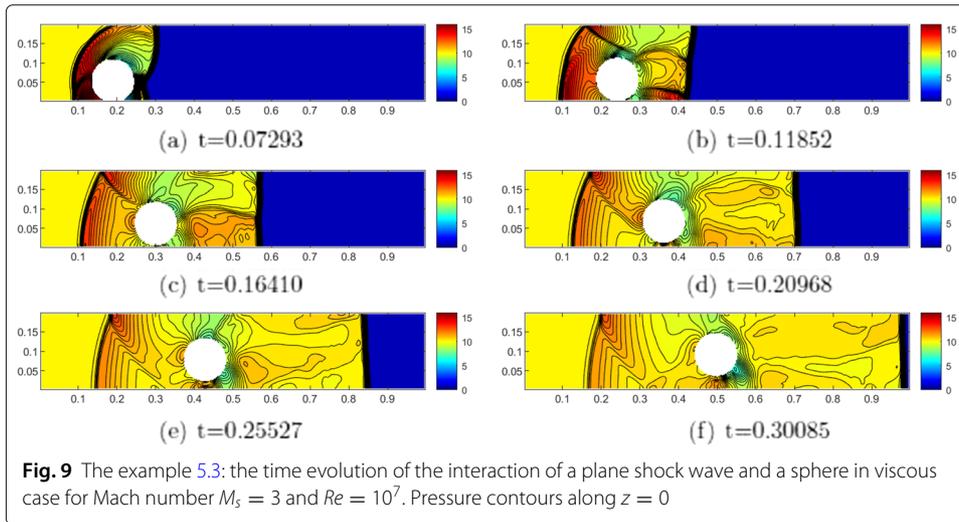


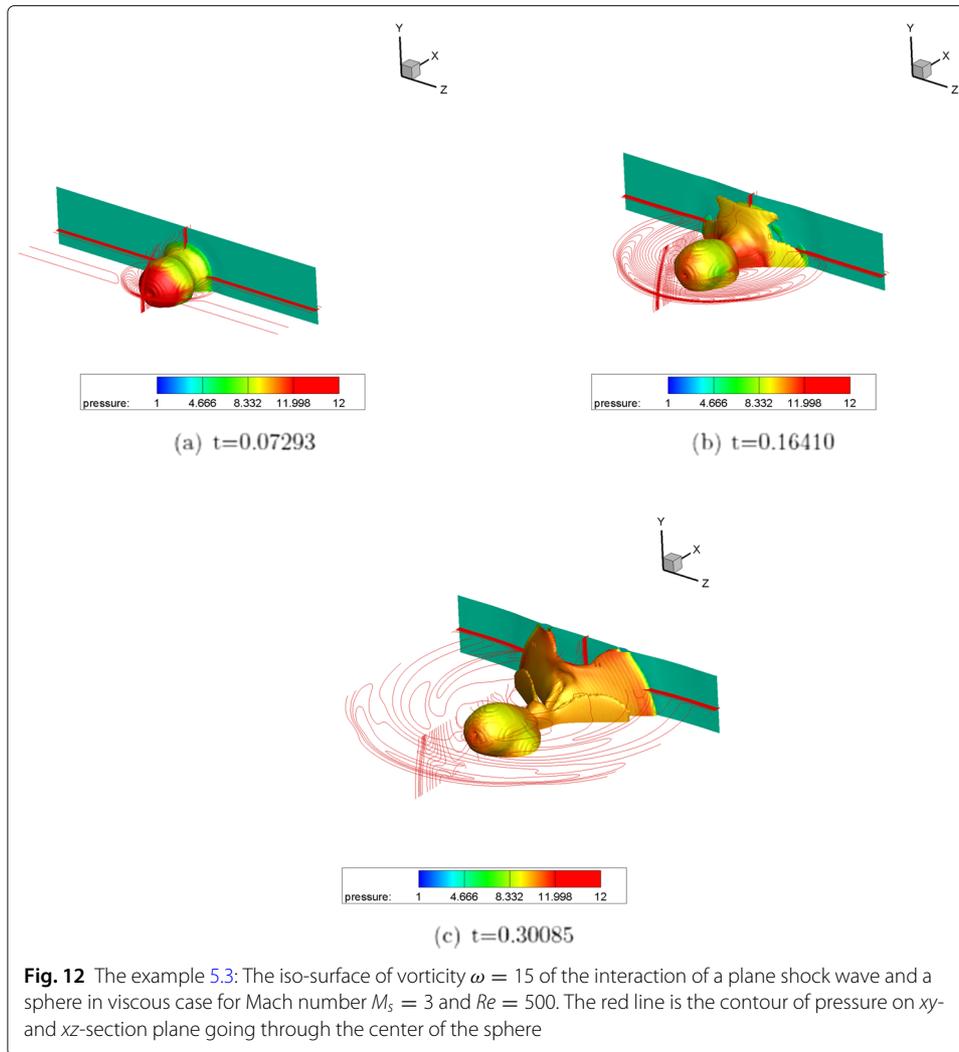
6 Conclusion

In this paper, we develop the high order numerical boundary treatments for compressible inviscid fluid and viscous fluid, and further use them to simulate the interaction between a complex moving rigid body and a shock wave in 3D. Schemes are based on the finite difference methods on fixed Cartesian meshes in complex moving geometries. Values on the ghost points can be obtained through a Taylor expansion on the boundary in the normal direction. And we can construct point values and normal derivatives via ILW procedure or extrapolation, or their combination according to the equations and boundary conditions. This is an extension of our works [13, 28], in which the boundary treatment was designed for 2D problems. Different from the 2D case, the local coordinate rotation required in the ILW procedure is not unique in 3D. We prove that the choice of the rotation will not affect results of the boundary treatment. Hence, we can choose a simple one in our simulation. On the other hand, the material derivative for inviscid fluid on the moving boundary with no-penetration condition is newly given. Consequently, we can simulate the body with both translation and rotation. In the numerical tests, we show the results of cylinder and sphere, indicating our algorithm is effective and robust.









Appendix A: WENO method

Numerical fluxes $\hat{\mathbf{F}}_{i+1/2,j,k}$, $\hat{\mathbf{G}}_{i,j+1/2,k}$ and $\hat{\mathbf{H}}_{i,j,k+1/2}$ can be approximated in a dimension-by-dimension fashion. This means the numerical flux $\hat{\mathbf{F}}_{i+1/2,j,k}$ is obtained by the one dimensional WENO reconstruction with j and k fixed. Likewise, the numerical flux $\hat{\mathbf{G}}_{i,j+1/2,k}$ (or $\hat{\mathbf{H}}_{i,j,k+1/2}$) is obtained by one dimensional approximation procedure with i and k (or i and j) fixed. In the following part, we give the details of the third order WENO procedure to construct $\hat{\mathbf{F}}_{i+1/2,j,k}$. And for ease of notation, we ignore the indexes j and k here.

1. Compute an average state $\mathbf{W}_{i+1/2}$, using either the simple arithmetic mean

$$\mathbf{W}_{i+1/2} = \frac{1}{2} (\mathbf{W}_i + \mathbf{W}_{i+1}),$$

or a Roe average satisfying

$$\mathbf{F}'(\mathbf{W}_{i+1/2})(\mathbf{W}_{i+1} - \mathbf{W}_i) = \mathbf{F}(\mathbf{W}_{i+1}) - \mathbf{F}(\mathbf{W}_i);$$

2. Compute the right eigenvectors, the left eigenvectors, and the eigenvalues of the Jacobian $\mathbf{F}'(\mathbf{W}_{i+1/2})$, and denote them by

$$\mathbf{R} = \mathbf{R}(\mathbf{W}_{i+1/2}), \quad \mathbf{R}^{-1} = \mathbf{R}^{-1}(\mathbf{W}_{i+1/2}), \quad \Lambda = \Lambda(\mathbf{W}_{i+1/2});$$

3. Transform all the point values \mathbf{W}_j and fluxes $\mathbf{F}_j = \mathbf{F}(\mathbf{W}_j)$ which are in the potential stencil of the WENO reconstruction to the local characteristic fields:

$$\tilde{\mathbf{W}}_j = \mathbf{R}^{-1}\mathbf{W}_j, \quad \tilde{\mathbf{F}}_j = \mathbf{R}^{-1}\mathbf{F}_j, \quad j = i - 1, \dots, i + 2;$$

4. Do the flux splitting:

$$\tilde{\mathbf{F}}_j^\pm = \frac{1}{2} \left(\tilde{\mathbf{F}}_j \pm \alpha \tilde{\mathbf{W}}_j \right), \quad j = i - 1, \dots, i + 2,$$

where, $\alpha = \max_l |\lambda_l(\mathbf{F}'(\mathbf{W}))|$ takes over the relevant range of \mathbf{W} , and λ_l are the eigenvalues of the Jacobian $\mathbf{F}'(\mathbf{W})$;

5. Perform the scalar WENO reconstruction procedure for each component of $\tilde{\mathbf{F}}^+ = (\tilde{f}_1^+, \dots, \tilde{f}_5^+)^T$ on the left-biased stencil to obtain the corresponding

component of the fluxes $\hat{\mathbf{F}}_{i+1/2}^+ = \left(\left(\hat{f}_1 \right)_{i+1/2}^+, \dots, \left(\hat{f}_5 \right)_{i+1/2}^+ \right)^T$. We summarize the procedure for each component $l = 1, \dots, 5$:

- (a) On the two small stencils $S^{(1)} = \{x_{i-1}, x_i\}$ and $S^{(2)} = \{x_i, x_{i+1}\}$, we have the low order approximation

$$\begin{aligned} \hat{f}_{i+1/2}^{(1)} &= -\frac{1}{2} \tilde{f}_{i-1} + \frac{3}{2} \tilde{f}_i, \\ \hat{f}_{i+1/2}^{(2)} &= \frac{1}{2} \tilde{f}_i + \frac{1}{2} \tilde{f}_{i+1}, \end{aligned}$$

and the linear weights

$$d_1 = \frac{1}{3}, \quad d_2 = \frac{2}{3};$$

- (b) Change the linear weights d_r to nonlinear weights ω_r :

$$\omega_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \quad \alpha_r = \frac{d_r}{(\beta_r + \epsilon)^2}, \quad r = 1, 2,$$

with $\epsilon = 10^{-6}$ to avoid the denominator being zero, and the smoothness indicators β_r on each small stencil are given as

$$\begin{aligned} \beta_1 &= \left(\tilde{f}_{i-1} - \tilde{f}_i \right)^2, \\ \beta_2 &= \left(\tilde{f}_i - \tilde{f}_{i+1} \right)^2; \end{aligned}$$

- (c) Find the third order reconstruction

$$\hat{f}_{i+1/2}^+ = \omega_1 \hat{f}_{i+1/2}^{(1)} + \omega_2 \hat{f}_{i+1/2}^{(2)};$$

6. Construct $\hat{\mathbf{F}}_{i+1/2}^-$ based on $\tilde{\mathbf{F}}^-$ and the right-biased stencil. The process to obtain $\hat{\mathbf{F}}_{i+1/2}^-$ is mirror-symmetric to that for $\hat{\mathbf{F}}_{i+1/2}^+$, with respect to the target point $x_{i+1/2}$;

7. Form the numerical flux as

$$\hat{\mathbf{F}}_{i+1/2} = \hat{\mathbf{F}}_{i+1/2}^+ + \hat{\mathbf{F}}_{i+1/2}^-;$$

8. Transform back into physical space

$$\hat{\mathbf{F}}_{i+1/2} = \mathbf{R} \hat{\mathbf{F}}_{i+1/2}.$$

Appendix B: A proof of Theorem 1

We prove the boundary treatment of the Euler equation here. The theorem also holds for the Navier-Stokes equation, and the proof is similar.

As before, assume $P_{i,j,k} = (x_i, y_j, z_k)$ is the ghost point we are concerned about, P_a is its pedal point at the boundary, and $\mathbf{n} = (n_1, n_2, n_3)$ is the normal vector which points from P_a to $P_{i,j,k}$. We choose $\mathbf{t}^1 = (t_1^1, t_2^1, t_3^1)$, $\mathbf{t}^2 = (t_1^2, t_2^2, t_3^2)$ as two orthogonal tangent vectors to take the rotation. The rotation matrix can be written as follows:

$$\mathbf{T} = \begin{pmatrix} n_1 & n_2 & n_3 \\ t_1^1 & t_2^1 & t_3^1 \\ t_1^2 & t_2^2 & t_3^2 \end{pmatrix}.$$

With this rotation, we have:

$$\begin{pmatrix} \hat{u}_{a,ext}^{(k)} \\ \hat{v}_{a,ext}^{(k)} \\ \hat{w}_{a,ext}^{(k)} \end{pmatrix} = \mathbf{T} \begin{pmatrix} u_{a,ext}^{(k)} \\ v_{a,ext}^{(k)} \\ w_{a,ext}^{(k)} \end{pmatrix}, \quad k = 0, 1, 2$$

where $u_{a,ext}^{(k)}, v_{a,ext}^{(k)}, w_{a,ext}^{(k)}$ can be obtained by extrapolation without using \mathbf{t}^1 and \mathbf{t}^2 . And the local characteristic decomposition can be written as:

$$\begin{pmatrix} (\hat{V}_1)_a^{(k)} \\ (\hat{V}_2)_a^{(k)} \\ (\hat{V}_3)_a^{(k)} \\ (\hat{V}_4)_a^{(k)} \\ (\hat{V}_5)_a^{(k)} \end{pmatrix} = \mathbf{L}_a \begin{pmatrix} \rho_{a,ext}^{(k)} \\ \hat{u}_{a,ext}^{(k)} \\ \hat{v}_{a,ext}^{(k)} \\ \hat{w}_{a,ext}^{(k)} \\ p_{a,ext}^{(k)} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2}\sqrt{\gamma p_{a,ext}^{(0)} \rho_{a,ext}^{(0)}} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -\frac{\rho_{a,ext}^{(0)}}{\sqrt{p_{a,ext}^{(0)}}} \\ 0 & \frac{1}{2}\sqrt{\gamma p_{a,ext}^{(0)} \rho_{a,ext}^{(0)}} & 0 & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} \rho_{a,ext}^{(k)} \\ \hat{u}_{a,ext}^{(k)} \\ \hat{v}_{a,ext}^{(k)} \\ \hat{w}_{a,ext}^{(k)} \\ p_{a,ext}^{(k)} \end{pmatrix}, \quad k = 0, 1, 2$$

Substitute the above characteristic variables into the right-hand-term of equation (27) and solve the linear system, we get:

$$\begin{cases} (\hat{U}_1)^{(0)} = \rho_{a,ext}^{(0)} \left(1 + \sqrt{\frac{\rho_{a,ext}^{(0)}}{\gamma p_{a,ext}^{(0)}}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right), \\ (\hat{U}_2)^{(0)} = \hat{u}_b - (n_1 u_{a,ext}^{(0)} + n_2 v_{a,ext}^{(0)} + n_3 w_{a,ext}^{(0)}) + (n_1 u_{a,ext}^{(0)} + n_2 v_{a,ext}^{(0)} + n_3 w_{a,ext}^{(0)}), \\ (\hat{U}_3)^{(0)} = t_1^1 u_{a,ext}^{(0)} + t_2^1 v_{a,ext}^{(0)} + t_3^1 w_{a,ext}^{(0)}, \\ (\hat{U}_4)^{(0)} = t_1^2 u_{a,ext}^{(0)} + t_2^2 v_{a,ext}^{(0)} + t_3^2 w_{a,ext}^{(0)}, \\ (\hat{U}_5)^{(0)} = p_{a,ext}^{(0)} + \sqrt{\gamma \rho_{a,ext}^{(0)} p_{a,ext}^{(0)}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b). \end{cases}$$

Apply the relation

$$\mathbf{U}^{(k)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & n_1 & t_1^1 & t_1^2 & 0 \\ 0 & n_2 & t_2^1 & t_2^2 & 0 \\ 0 & n_3 & t_3^1 & t_3^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \hat{\mathbf{U}}^{(k)}, \quad k = 0, 1, 2. \tag{45}$$

We have

$$\begin{cases} (U_1)^{(0)} = \rho_{a,ext}^{(0)} \left(1 + \sqrt{\frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right), \\ (U_2)^{(0)} = n_1 \left(\hat{u}_b - (n_1 u_{a,ext}^{(0)} + n_2 v_{a,ext}^{(0)} + n_3 w_{a,ext}^{(0)}) \right) + u_{a,ext}^{(0)}, \\ (U_3)^{(0)} = n_2 \left(\hat{u}_b - (n_1 u_{a,ext}^{(0)} + n_2 v_{a,ext}^{(0)} + n_3 w_{a,ext}^{(0)}) \right) + v_{a,ext}^{(0)}, \\ (U_4)^{(0)} = n_3 \left(\hat{u}_b - (n_1 u_{a,ext}^{(0)} + n_2 v_{a,ext}^{(0)} + n_3 w_{a,ext}^{(0)}) \right) + w_{a,ext}^{(0)}, \\ (U_5)^{(0)} = p_{a,ext}^{(0)} + \sqrt{\gamma \rho_{a,ext}^{(0)} P_{a,ext}^{(0)}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b). \end{cases}$$

Here, the orthogonality of rotation matrix T is used. And we can see that all the terms here are independent of \mathbf{t}^1 and \mathbf{t}^2 .

Similarly, substitute the characteristic variables into equation (28) and solve the linear system, we get:

$$\begin{cases} (\hat{U}_1)^{(1)} = \rho_{a,ext}^{(1)} + \rho_{a,ext}^{(0)} \left(\frac{\rho_{a,ext}^{(0)} \frac{D\hat{u}}{Dt} - p_{a,ext}^{(1)}}{\gamma p_{a,ext}^{(0)}} + \frac{D\hat{u}}{Dt} \left(\frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}} \right)^{\frac{3}{2}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right), \\ (\hat{U}_2)^{(1)} = n_1 u_{a,ext}^{(1)} + n_2 v_{a,ext}^{(1)} + n_3 w_{a,ext}^{(1)} + \hat{u}_{res}, \\ (\hat{U}_3)^{(1)} = t_1^1 u_{a,ext}^{(1)} + t_2^1 v_{a,ext}^{(1)} + t_3^1 w_{a,ext}^{(1)}, \\ (\hat{U}_4)^{(1)} = t_1^2 u_{a,ext}^{(1)} + t_2^2 v_{a,ext}^{(1)} + t_3^2 w_{a,ext}^{(1)}, \\ (\hat{U}_5)^{(1)} = \rho_{a,ext}^{(0)} \frac{D\hat{u}}{Dt} \left(1 + \sqrt{\frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right), \end{cases}$$

where

$$\frac{D\hat{u}}{Dt} = \hat{\mathbf{u}} \cdot \frac{D\hat{\mathbf{n}}}{Dt} - \frac{d}{dt} (\mathbf{V}_b \cdot \mathbf{n})$$

and

$$\hat{u}_{res} = -\frac{\rho_{a,ext}^{(0)} \frac{D\hat{u}}{Dt} - p_{a,ext}^{(1)}}{\sqrt{\gamma P_{a,ext}^{(0)} \rho_{a,ext}^{(0)}}} + \frac{D\hat{u}}{Dt} \frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b)$$

are terms which are independent of \mathbf{t}^1 and \mathbf{t}^2 .

Apply the relation (45), we get

$$\begin{cases} (U_1)^{(1)} = \rho_{a,ext}^{(1)} + \rho_{a,ext}^{(0)} \left(\frac{\rho_{a,ext}^{(0)} \frac{D\hat{u}}{Dt} - p_{a,ext}^{(1)}}{\gamma P_{a,ext}^{(0)}} + \frac{D\hat{u}}{Dt} \left(\frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}} \right)^{\frac{3}{2}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right), \\ (U_2)^{(1)} = n_1 \hat{u}_{res} + u_{a,ext}^{(1)}, \\ (U_3)^{(1)} = n_2 \hat{u}_{res} + v_{a,ext}^{(1)}, \\ (U_4)^{(1)} = n_3 \hat{u}_{res} + w_{a,ext}^{(1)}, \\ (U_5)^{(1)} = \rho_{a,ext}^{(0)} \frac{D\hat{u}}{Dt} \left(1 + \sqrt{\frac{\rho_{a,ext}^{(0)}}{\gamma P_{a,ext}^{(0)}}} (\hat{u}_{a,ext}^{(0)} - \hat{u}_b) \right). \end{cases}$$

Again, we can see the expression of $U_a^{(1)}$ does not explicitly contain \mathbf{t}^1 and \mathbf{t}^2 . Also, the second order derivative $U_a^{(2)}$, which is obtained by extrapolation directly, can be got without using the tangent vectors.

Therefore, the ghost points can be defined by a Taylor expansion without tangent vectors.

Appendix C: WENO extrapolation

The idea of WENO extrapolation is using nonlinear weights to combine the approximations on candidate substencils.

Firstly, we select three candidate substencils S_0, S_1, S_2 for extrapolation. There is more than one way to select them. Thanks to the regular boundaries like a sphere, we can simplify the traditional point finding method in [17]. Here, the inner lattice points whose distance from point P_a is less than $1.1h, 2.1h$ and $3.1h$ are grouped to compose S_0, S_1, S_2 respectively, where $h = \max(\Delta x, \Delta y, \Delta z)$ is the largest mesh size.

Then, for each component of \mathbf{V} , we reconstruct a polynomial of $p_m(x, y, z)$ ($m = 0, 1, 2$) with the least square method on each candidate stencil S_m , respectively. Values of $\frac{\partial^k V_i}{\partial \hat{x}^k}$ ($i = 1, 2, 3, 4$) can be extrapolated as:

$$\frac{\partial^k V_i}{\partial \hat{x}^k} = \sum_{m=0}^2 d_m \frac{\partial^k}{\partial \hat{x}^k} p_m|_{(x,y,z)=(x_{(a)},y_{(a)},z_{(a)})} \tag{46}$$

where $d_0 = \Delta x^2 + \Delta y^2 + \Delta z^2, d_1 = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}, d_2 = 1 - d_0 - d_1$.

Next, we change the approximation (46) to a WENO type extrapolation to avoid spurious oscillation, with the following form:

$$\frac{\partial^k V_i}{\partial \hat{x}^k} = \sum_{m=0}^2 \omega_m \frac{\partial^k}{\partial \hat{x}^k} p_m|_{(x,y,z)=(x_{(a)},y_{(a)},z_{(a)})}, \tag{47}$$

where $\omega_0, \omega_1, \omega_2$ are nonlinear weights depending on the smoothness of the functions on candidate substencils, taken as

$$\omega_m = \frac{\alpha_m}{\sum_{s=0}^2 \alpha_s}, \quad m = 0, 1, 2, \tag{48}$$

with

$$\alpha_m = \frac{d_m}{(\epsilon + \beta_m)^2}. \tag{49}$$

The small number ϵ is taken as 10^{-6} to avoid zero denominator. The smoothness indicators β_m are similar to those in [31]

$$\beta_0 = \Delta x^2 + \Delta y^2 + \Delta z^2, \tag{50}$$

$$\beta_m = \sum_{1 \leq |\mu| \leq m} \iiint_K |K|^{2/3|\mu|-1} (D^\mu p_m)^2 dx dy dz$$

where μ is a multi-index and $K = [x_{(a)} - \Delta x/2, x_{(a)} + \Delta x/2] \times [y_{(a)} - \Delta y/2, y_{(a)} + \Delta y/2] \times [z_{(a)} - \Delta z/2, z_{(a)} + \Delta z/2]$. With the careful design of nonlinear weights ω_m , the WENO extrapolation can meet third order accuracy when the function is smooth enough on all candidate substencils.

Acknowledgments

N/A.

Authors' contributions

All the authors contributed to and approved this manuscript.

Funding

This work has been supported by the National Numerical Windtunnel project (No. NNW2019ZT4-B10), and the National Natural Science Foundation of China (Nos. 11901555, 11901213, 11871448, 11732016).

Availability of data and materials

The data and materials in this work are available from the corresponding author on reasonable request.

Declarations

Competing interests

The authors declare that they have no competing interests.

Author details

¹School of Mathematical Sciences, University of Science and Technology of China, 230026 Hefei, Anhui, China. ²School of Mathematics, Hefei University of Technology, 230000 Hefei, Anhui, China. ³School of Mathematics, South China University of Technology, 510641 Guangzhou, Guangdong, China. ⁴State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, 621000 Mianyang, Sichuan, China.

Received: 31 August 2021 Accepted: 28 October 2021

Published online: 21 February 2022

References

1. Berger MJ, Helzel C, Leveque RJ (2003) h-box methods for the approximation of hyperbolic conservation laws on irregular grids. *SIAM J Numer Anal* 41:893–918
2. Baeza A, Mulet P, Zoro D (2016) High order accurate extrapolation technique for finite difference methods on complex domains with Cartesian meshes. *J Sci Comput* 66:761–791
3. Kreiss H-O, Petersson NA (2006) A second order accurate embedded boundary method for the wave equation with Dirichlet data. *SIAM J Sci Comput* 27:1141–1167
4. Kreiss H-O, Petersson NA, Yström J (2002) Difference approximations for the second order wave equation. *SIAM J Numer Anal* 40:1940–1967
5. Kreiss H-O, Petersson NA, Yström J (2004) Difference approximations of the Neumann problem for the second order wave equation. *SIAM J Numer Anal* 42:1292–1323
6. Nilsson S, Petersson NA, Sjögreen B, Kreiss H-O (2007) Stable difference approximations for the elastic wave equation in second order formulation. *SIAM J Numer Anal* 45:1902–1936
7. Sjögreen B, Petersson NA (2007) A Cartesian embedded boundary method for hyperbolic conservation laws. *Commun Comput Phys* 2:1199–1219
8. Mittal R, Iaccarino G (2005) Immersed boundary methods. *Ann Rev Fluid Mech* 37:239–261
9. Peskin CS (2002) The immersed boundary method. *Acta Numerica* 11:1–39
10. Qu Y, Shi R, Batra RC (2018) An immersed boundary formulation for simulating high-speed compressible viscous flows with moving solids. *J Comput Phys* 354:672–691
11. Vanna FD, Picano F, Benini E (2020) A sharp-interface immersed boundary method for moving objects in compressible viscous flows. *Comput Fluids* 201:104415
12. Wang J, Gu X, Wu J (2020) A sharp-interface immersed boundary method for simulating high-speed compressible inviscid flows. *Adv Appl Math Mech* 12:545–563
13. Cheng Z, Liu S, Jiang Y, Lu J, Zhang M, Zhang S (2021) A high order boundary scheme to simulate a complex moving rigid body under the impingement of a shock wave. *Appl Math Mech (Engl Ed)* 42:841–854
14. Ding S, Shu C-W, Zhang M (2020) On the conservation of finite difference WENO schemes in non-rectangular domains using the inverse Lax-Wendroff boundary treatments. *J Comput Phys* 415:109516
15. Lu J, Fang J, Tan S, Shu C-W, Zhang M (2016) Inverse Lax-Wendroff procedure for numerical boundary conditions of convection-diffusion equations. *J Comput Phys* 317:276–300
16. Lu J, Shu C-W, Tan S, Zhang M (2021) An inverse Lax-Wendroff procedure for hyperbolic conservation laws with changing wind direction on the boundary. *J Comput Phys* 426:109940
17. Tan S, Shu C-W (2010) Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *J Comput Phys* 229:8144–8166
18. Tan S, Shu C-W (2011) A high order moving boundary treatment for compressible inviscid flows. *J Comput Phys* 230:6023–6036
19. Tan S, Shu C-W (2013) Inverse Lax-Wendroff procedure for numerical boundary conditions of hyperbolic equations: survey and new developments. In: Melnik R, Kotsireas I (eds). *Advances in Applied Mathematics, Modeling and Computational Science*. Fields Institute Communications 66, Springer, New York, pp 41–63
20. Tan S, Wang C, Shu C-W, Ning J (2012) Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. *J Comput Phys* 231:2510–2527
21. Huang L, Shu C-W, Zhang M (2008) Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation. *J Comput Math* 26:336–346
22. Xiong T, Zhang M, Zhang Y-T, Shu C-W (2010) Fifth order fast sweeping WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment. *J Sci Comput* 45:514–536
23. Filbet F, Yang C (2013) An inverse Lax-Wendroff method for boundary conditions applied to Boltzmann type models. *J Comput Phys* 245:43–61
24. Li T, Shu C-W, Zhang M (2017) Stability analysis of the inverse Lax-Wendroff boundary treatment for high order central difference schemes for diffusion equations. *J Sci Comput* 70:576–607
25. Li T, Lu J, Shu C-W (2021) Stability analysis of inverse Lax-Wendroff boundary treatment of high order compact difference schemes for parabolic equations. *J Comput Appl Math* 400:113711
26. Li T, Shu C-W, Zhang M (2016) Stability analysis of the inverse Lax-Wendroff boundary treatment for high order upwind-biased finite difference schemes. *J Comput Appl Math* 299:140–158
27. Vilar F, Shu C-W (2015) Development and stability analysis of the inverse Lax-Wendroff boundary treatment for central compact schemes. *ESAIM Math Model Numer Anal (M²AN)* 49:39–67

28. Liu S, Jiang Y, Shu C-W, Zhang M, Zhang S (2021) A high order moving boundary treatment for convection-diffusion equations. <https://www.brown.edu/research/projects/scientific-computing/sites/brown.edu.research.projects.scientific-computing/files/uploads/A%20high%20order%20moving%20boundary%20treatment.pdf>
29. Jiang G-S, Shu C-W (1996) Efficient implementation of weighted ENO schemes. *J Comput Phys* 126:202–228
30. Shu C-W, Osher S (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J Comput Phys* 77:439–471
31. Zhang Y-T, Shu C-W (2009) Third order WENO scheme on three dimensional tetrahedral meshes. *Commun Comput Phys* 5:836–848

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

