Journal of Computational Physics 326 (2016) 862–877

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Cartesian grid method for gas kinetic scheme on irregular geometries

Songze Chen [a,*], Kun Xu [b], Zhihui Li [c]

[a] Huazhong University of Science and Technology, State Key Laboratory of Coal Combustion, Luoyu Road 1037, Wuhan, China
[b] The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China
[c] Hypervelocity Aerodynamics Institute, China Aerodynamics Research and Development Center, Mianyang 621000, China

ARTICLE INFO

Article history:
Received 15 February 2015
Received in revised form 1 May 2016
Accepted 9 September 2016
Available online 20 September 2016

Keywords:
Gas kinetic scheme
Cartesian grid
Viscous flow
Supersonic flow

ABSTRACT

A Cartesian grid method combined with a simplified gas kinetic scheme is presented for subsonic and supersonic viscous flow simulation on complex geometries. Under the Cartesian mesh, the boundaries are represented by a set of direction-oriented boundary points, and the computational grid points are classified into four different categories, the fluid point, the solid point, the drop point, and the interpolation point. A constrained weighted least square method is employed to evaluate the physical quantities at the interpolation points. Different boundary conditions, including isothermal boundary, adiabatic boundary, and Euler slip boundary, are presented by different interpolation strategies. We adopt a simplified gas kinetic scheme as the flux solver for both subsonic and supersonic flow computations. The methodology of constructing a simplified kinetic flux function can be extended to other flow systems. A few numerical examples are used to validate the Cartesian grid method and the simplified flux solver. The reconstruction scheme for recovering the boundary conditions of compressible viscous and heat conducting flow with a Cartesian mesh can provide a smooth distribution of physical quantities at solid boundary, and present an accurate solution for the flow study with complex geometry.

1. Introduction

Solving flow problem on conformal mesh or unstructured mesh is the most practical strategy in industry applications. However, the grid generation on complex geometry is time consuming and requires sophisticated technique. In this sense, the grid generation on complex computational domain is still the bottle neck of practical applications. Thereby, the Cartesian grid method arises, in which the flow region is discretized by a Cartesian grid regardless of the shapes of objects inside the flow region. The obvious advantage of this method over the conventional conformal approach is that the computational mesh is easily generated when different geometries are considered. Cartesian grid methods free the researchers and engineers from the burdensome grid generation, but introduce two new problems about boundary treatment.

The first problem with the Cartesian grid is about how to represent the boundary. It depends on the boundary property. For the multi-fluid and multi-phase flows, where the boundaries or the interfaces are deformable, the volume-of-fluid (VOF) method [1] and the phase-field approach [2] are the most popular approaches. The boundary is reconstructed from an

[*] Corresponding author.
E-mail address: jacksongze@hust.edu.cn (S. Chen).

http://dx.doi.org/10.1016/j.jcp.2016.09.018
0021-9991/© 2016 Elsevier Inc. All rights reserved.

Nomenclature

$\mathbf{a}$	Expansion coefficient of the spatial derivative of g
$\mathbf{F}$	Macroscopic fluxes for density, momentum, and energy
$\mathbf{n}$	Normal direction of a surface
$\mathbf{t}$	Tangential direction of a surface
$\mathbf{U}$	Macroscopic velocity
$\mathbf{u}$	Particle velocity
$\mathbf{W}$	Conserved macroscopic variables, including density, momentum, and energy
ρ	Gas density
τ	Relaxation time of kinetic equation
f	Particle velocity distribution function
g	Equilibrium state of particle velocity distribution function
K	Equivalent internal degree of freedom which is 1 for monatomic gas, and 3 for diatomic gas in two dimensional flow
M	The number of the boundary points which are chosen to be the constraint
N	The number of the fluid points in the boundary interpolation stencil
P	Interpolation polynomial
p	Gas pressure
R	Gas constant
T	Gas temperature
w	Weight function
Ma	Mach number
Re	Reynolds number

auxiliary function, say, a marker function. Or, the boundary is captured by solving a partial differential equation for the phase field. These approaches are efficient for boundary deformation and splitting. But obviously, this kind of approaches is less accurate than the Lagrangian representation of the boundary. The latter approach takes the boundary as a sharp interface either explicitly tracking as curves [3] or as level sets [4]. Sharp interface is desirable for high Reynolds number flows where the boundary layer plays an important role.

The second problem is how to impose the boundary condition. Following the classification by Mittal and Iaccarino [5], there are two different categories. The first one is the continuous forcing approach; the second one is the discrete forcing approach. The continuous forcing approach directly modifies the governing equation by adding a forcing term to take the boundary effect into account [6,7]. An ideal force term is represented by a Dirac delta function. Since the boundary cuts the grid line at arbitrary location, the forcing should be distributed over a band of cells around the boundary point. This approach results in a diffusive boundary. The discrete forcing approach imposes the boundary condition on the numerical solution directly at discrete level. Mittal and Iaccarino [5] further subdivided the second category into “Indirect BC Imposition” and “Direct BC Imposition”. The former one employs a forcing term which is determined from a priori estimation of flow field at discrete level [8]. The external force is explicitly computed in advance or solved by an implicit method to guarantee the no-slip boundary condition [9,10]. However, the artificial forcing procedure diffuses the flow field around the boundary. For this reason, the “Direct BC Imposition” approaches retain the boundary as a sharp interface with no spreading. This can usually be accomplished by modifying the computational stencil near the boundary to directly impose the boundary condition. This kind of approach is always referred to as Cartesian grid method which is the main focus of this study.

Berger and LeVeque [11] presented a Cartesian grid algorithm with adaptive refinement to compute flows around arbitrary geometries by solving the Euler equations. They treated the intersection between the grid line and the boundaries as a grid point and performed conventional finite difference scheme with amendments of boundary fluxes. However, the small cell instability was observed. Pember et al. [12] proposed a corrector applied to cells at the boundary to redistribute flow quantities in order to maintain the conservation, therefore, avoid time step restrictions arising from small cells. In these methods, the amendments or the correctors depend on the specific physical system, thereby, the numerical scheme is difficult for further extension.

Ye et al. [13] and Udaykumar et al. [14] proposed a sharp interface Cartesian grid method for solving the incompressible Navier–Stokes equations with complex moving boundaries. The boundaries cut the mesh, then form some trapezoidal cells. If the trapezoidal cell is very small, then it merges with its neighboring cell, and forms an irregular control volume. The finite volume method can be applied on the merging control volume. Then the small cell problem is circumvented. They proposed an interpolation procedure to construct the flow field near the boundary, so that the boundary condition is satisfied. Tullio et al. [15] and Palma et al. [16] combines the method for solving the three-dimensional preconditioned Navier–Stokes equations for compressible flows with an immersed boundary approach, to provide a Cartesian-grid method for computing complex flows over a wide range of the Mach number. Moreover, a flexible local grid refinement technique is employed to achieve high resolution near the immersed body and in other high-flow-gradient regions.

Forrer and Jeltsch [17] proposed a ghost cell method to apply the boundary condition on Cartesian grid for inviscid compressible Euler equations. The flow quantities at ghost cells which locate inside the boundary are set by their images in the fluid side across the boundary. As a result, the grid points near the boundary also preserve a complete control volume. Thus the small cell restriction is removed. Hereafter, this kind of method is referred to as the boundary interpolation method. In recent years, the boundary interpolation method receives more and more attentions. Udaykumar et al. [3] proposed a finite-difference formulation to track solid-liquid boundaries on a fixed underlying grid. The interface is not of finite thickness but is treated as discontinuity and is explicitly tracked. The imposition of boundary conditions exactly on a sharp interface is performed using simple stencil readjustments in the vicinity of the interface. Ghias et al. [18] proposed an immersed boundary method for computing viscous, subsonic compressible flows with complex shaped stationary immersed boundaries. The method also employs a ghost cell technique for imposing the boundary conditions on the immersed boundaries. As Gibou et al. [19] pointed out that the single directional interpolation scheme is poorly behaved for small cells. Therefore, Ghias et al. turned to a multidimensional bilinear interpolation and evaluated the flow variables at ghost cell to avoid the small cell problem.

Actually, many Cartesian grid method solving the compressible Navier–Stokes equations are based on the interpolation method to handle the boundary conditions. Peng et al. [20] has compared the forcing method and the interpolation method in the framework of lattice Boltzmann method. They concluded that the interpolation method perform better than the forcing method. In recent years, more delicate interpolation procedures are developed for boundary treatment. Seo and Mittal [21] proposed a new sharp-interface immersed boundary method for the computation of low-Mach number flow-induced acoustic flow around complex geometries. Their method implemented the boundary condition on the immersed boundary to a high order by combining the ghost cell approach with a weighted least square method based on a high order approximating polynomial. Duan et al. [22] extended the Cartesian grid method to high order accuracy, and simulated turbulent flow at high Mach numbers.

In this paper, we are going to follow the study of Seo and Mittal [21] to further develop the boundary interpolation method on Cartesian grid, and combine it with well developed gas kinetic schemes. The main purpose is to formulate a constrained weighted least square [23] interpolation for different boundary condition.

2. Simplified gas kinetic scheme

For the sake of completeness, we will briefly introduce the simplified gas kinetic scheme. The gas kinetic schemes (GKS) were developed by Xu et al. [24–28]. It is based on kinetic theory, using particle distribution function to simulate macroscopic flow motion. Gas kinetic scheme can be considered as a flux solver of a generalized Riemann problem. In this study, the two dimensional Bhatnagar–Gross–Krook kinetic equation [29] is employed. It takes the following form,

$$\frac{\partial f}{\partial t} + \mathbf{u} \cdot \frac{\partial f}{\partial \mathbf{x}} = \frac{g - f}{\tau}, \tag{1}$$

where $f(\mathbf{x}, t, \mathbf{u}, \xi)$ represents the particle velocity distribution function. It is a function of the location $\mathbf{x}$, the particle velocity $\mathbf{u}$, the internal degree of freedom ξ and the time t. The right hand side of Eq. (1) represents the relaxation process. g is the equilibrium state which reads,

$$g = \mathcal{M}[< f >] \equiv \rho \left(\frac{1}{2\pi RT} \right)^{\frac{K+2}{2}} e^{-\frac{1}{2RT}(\mathbf{u}-\mathbf{U})^2}, \tag{2}$$

where ρ denotes density, $\mathbf{U}$ represents macroscopic velocity, T denotes temperature, R is gas constant, and K is equivalent internal degree of freedom which is 1 for monatomic gas, and 3 for diatomic gas in two dimensional simulation. The conserved variables $\mathbf{W}$ can be derived by taking moments of f,

$$\mathbf{W} = \begin{pmatrix} \rho \\ \rho \mathbf{U} \\ \rho E \end{pmatrix} = \int \psi f d\mathbf{u} d\xi =< f >,$$

where $\psi = (1, \mathbf{u}, \frac{1}{2}(\mathbf{u}^2 + \xi^2))^T$, and E denotes the energy per unit mass. The macroscopic flux of a given distribution function f can be written as,

$$\mathbf{F} = \int \psi \mathbf{n} \cdot \mathbf{u} f d\mathbf{u} d\xi =< \mathbf{n} \cdot \mathbf{u} f >,$$

where $\mathbf{n}$ denotes a certain direction, say, the normal direction of a surface.

The original GKS is a finite volume method. If the numerical flux $\mathbf{F}$ is known at the cell interfaces of a cell, the macroscopic variables $\mathbf{W}^{n+1}$ for the next time step becomes,

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} - \frac{1}{V} \int\limits_{t^n}^{t^{n+1}} \sum_k \mathbf{F}_k S_k dt, \tag{3}$$

where S_k denotes the area of the kth surface of the cell, and V is the volume of the cell. In this study, we adopt the finite difference formulation. Then, for two dimensional problems,

$$\mathbf{W}_{i,j}^{n+1} = \mathbf{W}_{i,j}^{n} - \int\limits_{t^n}^{t^{n+1}} (\frac{\mathbf{F}_{i+1/2,j} - \mathbf{F}_{i-1/2,j}}{\Delta x} + \frac{\mathbf{F}_{i,j+1/2} - \mathbf{F}_{i,j-1/2}}{\Delta y}) dt. \tag{4}$$

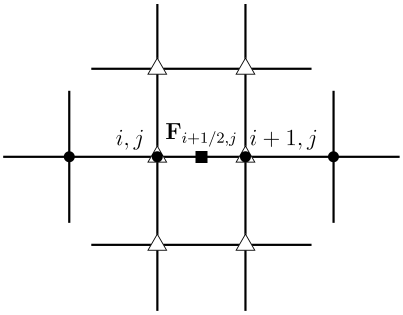

Fig. 1. The stencil for numerical flux in x direction.

To calculate the numerical flux, the distribution function at the flux point is needed. We adopt the simplified GKS [24,25], in which the modified Chapman–Enskog expansion is employed as the local distribution function, to evaluate the numerical flux at the flux point,

$$f(t) = (1-\alpha) f_0 + \alpha g_c - \tau(\mathbf{u}\cdot g_{\mathbf{x}} + g_t) + g_t t. \tag{5}$$

Here g_c is the equilibrium state at the cell interface. It is constructed by central difference scheme. The subscript c denotes the central difference scheme. f_0 is a non-equilibrium distribution function accounting for the upwind effect. In this paper, $\alpha = 0$, that is,

$$f(t) = f_0 - \tau(\mathbf{u}\cdot g_{\mathbf{x}} + g_t) + g_t t. \tag{6}$$

Fig. 1 shows the flux stencil used in 2D reconstruction. The solid square ($\mathbf{F}_{i+1/2,j}$) represents the flux point. The solid circles represent the upwind stencil, and the empty triangles represent the central difference stencil. A uniform mesh is assumed. The non-equilibrium state f_0 is

$$f_0 = \begin{cases} g_l = \mathcal{M}[\mathbf{W}_l], & u > 0 \\ g_r = \mathcal{M}[\mathbf{W}_r], & u \le 0 \end{cases}, \tag{7}$$

where the subscript ‘l’ and ‘r’ represent the left and right sides of the flux point respectively. $\mathbf{W}_l$ and $\mathbf{W}_r$ are constructed along the grid line. For the fluid side, the 3rd order WENO reconstruction is adopted,

$$W_l = \frac{w_{-1}W_{-1} + w_0 W_0}{w_{-1} + w_0}, \qquad W_r = \frac{w_0 W_0 + w_1 W_1}{w_0 + w_1}, \tag{8}$$

where W represents every single macroscopic quantity of $\mathbf{W}$. The weights are given as follows,

$$w_{-1} = \frac{1}{4(s_{i-1}^2 + \varepsilon)}, \quad w_0 = \frac{3}{4(s_i^2 + \varepsilon)}, \quad w_1 = \frac{1}{4(s_{i+1}^2 + \varepsilon)}, \tag{9}$$

where ε is a small positive number to avoid zero denominator, and $s_i = W_{i+1,j} - W_{i,j}$,

$$W_{-1} = \frac{3}{2}W_{i,j} - \frac{1}{2}W_{i-1,j},\ W_0 = \frac{1}{2}W_{i,j} + \frac{1}{2}W_{i+1,j},\ W_1 = \frac{3}{2}W_{i+1,j} - \frac{1}{2}W_{i+2,j}. \tag{10}$$

The spatial derivative in the second term of right hand side of Eq. (6) can be expressed in the expansion of equilibrium state, which formally reads,

$$g_{\mathbf{x}} = \mathbf{a}\cdot\psi\mathcal{M}[\mathbf{W}^n], \tag{11}$$

where the expansion coefficient $\mathbf{a}$ reads,

$$\mathbf{a}\cdot\psi = a_{0,d} + \mathbf{a}_{1,d}\cdot\mathbf{u} + a_{2,d}\frac{\mathbf{u}^2 + \xi^2}{2}, \quad d = 1,\ldots,D, \tag{12}$$

where $\mathbf{a}_{1,d} = (a_{1,d1},\ldots,a_{1,dD})$, and D denotes the dimension. For example, for two dimensional problems, $D = 2$. Then take the conservative moments of the spatial derivatives of equilibrium state, we have,

$$\mathbf{a}\cdot < \psi\mathcal{M}[\mathbf{W}^n] >= \mathbf{W}_{\mathbf{x}}, \tag{13}$$

where $\mathbf{W}^n$ denotes the conservative variables at the beginning of the time step. So, the corresponding conservative variables are $\mathbf{W}^n =< f_0 >$ for the Eq. (6). Then $\mathbf{W}_{\mathbf{x}}$ is derived by a central difference scheme. Take two dimensional problems as an example,

$$\mathbf{W}_x = \frac{\mathbf{W}_{i+1,j} - \mathbf{W}_{i,j}}{\Delta x} \tag{14}$$

$$\mathbf{W}_y = \frac{1}{4\Delta y}(\mathbf{W}_{i,j+1} + \mathbf{W}_{i+1,j+1} - \mathbf{W}_{i,j-1} - \mathbf{W}_{i+1,j-1}) \tag{15}$$

Solving the Eq. (13), we get the expression of $g_{\mathbf{x}}$.

The temporal derivative is estimated by the direct difference of the equilibrium state on the time interval.

$$g_t = \frac{1}{\Delta t}(\mathcal{M}[\mathbf{W}^{n+1}] - \mathcal{M}[\mathbf{W}^n]), \tag{16}$$

where $\mathbf{W}^{n+1}$ denotes the conservative variables at the end of the time step. It can be derived by applying the conservation constraint, $< \tau(g_t + \mathbf{u} \cdot g_{\mathbf{x}}) >= 0$ at a cell interface, that is,

$$\mathbf{W}^{n+1} = \mathbf{W}^n - \Delta t < \mathbf{u} \cdot g_{\mathbf{x}} > . \tag{17}$$

The time dependent distribution function at flux point is,

$$f(t) = f_0 - \tau(\mathbf{u} \cdot g_{\mathbf{x}} + \frac{\mathcal{M}[\mathbf{W}^{n+1}] - \mathcal{M}[\mathbf{W}^n]}{\Delta t}) + t\frac{\mathcal{M}[\mathbf{W}^{n+1}] - \mathcal{M}[\mathbf{W}^n]}{\Delta t}$$

Therefore, the numerical flux reads,

$$\begin{aligned} \int_{t^n}^{t^{n+1}} \mathbf{F} dt &= \mathbf{n} \cdot \int_{t^n}^{t^{n+1}} \int \psi \mathbf{u} f(t) d\mathbf{u} d\xi dt \\ &= \mathbf{n} \cdot (\Delta t < \mathbf{u} f_0 > -\tau \Delta t < \mathbf{u}\mathbf{u} \cdot g_{\mathbf{x}} > \\ &\quad + (\frac{\Delta t}{2} - \tau) < \mathbf{u}(\mathcal{M}[\mathbf{W}^{n+1}] - \mathcal{M}[\mathbf{W}^n]) >). \end{aligned} \tag{18}$$

Up to this point, the simplified gas kinetic scheme is completed. However, if the flux points are close to the boundary, the grid points in the flux stencil may be invalid, since these points may locate inside solid wall or too close to the wall. A boundary interpolation procedure must be invoked to reconstruct the quantities at these points.

3. Cartesian grid method

In the past years, curvilinear grid [30,31] and unstructured grid [32,33] have been employed to handle the complex geometries under the framework of kinetic schemes. Several immersed boundary methods for gas kinetic scheme have also been proposed for incompressible flows [34,35] in recent years. In this study, sharp interface is retained, and a constrained weighted least square method is employed for the boundary interpolation procedure.

Fig. 2 illustrates the stencil for the boundary interpolation. The circles represent the fluid points; the squares represent the drop points which are excluded from the flux stencil and the interpolation stencil; the diamonds represent the solid points, while the empty diamonds are the interpolation points or ghost points; and the boundary points are denoted by their location and normal directions. The right top part of Fig. 2 illustrates how to classify a grid point. The grid points (O, D, C) are solid points, since they locate at the back of the neighboring boundary points (E and F). The grid points A and B are fluid points, which locate at the front of boundary points (E or F). If a grid point is too close to the boundary point, then, it is marked as a drop point. When a solid point or a drop point is involved in a flux stencil, then it becomes an interpolation point if the distance to the boundary along the upwind stencil is larger than half cell size. After defining different kind of grid points, all the flux stencils are composed of interpolation points and fluid points. Meanwhile, the quantities at the drop points are passively interpolated from fluid points. Therefore, the small cells formed by the cutting process are totally avoided, and no stability problem is observed.

The boundary condition is satisfied by assigning the corresponding values at the interpolation points or ghost points. The solid points which are involved in a flux stencil are employed as interpolation points. To satisfy the boundary condition, the fluid variables at the interpolation point should be assigned by appropriate interpolation. As shown in Fig. 2, we use the fluid points inside the big circle to reconstruct the fluid variables at interpolation points to enforce the corresponding boundary conditions.

3.1. Constrained weighted least square interpolation

Seo [21] proposed a weighted least square interpolation for low Mach number acoustic simulation. For low Mach number flow, the boundary value can be predetermined, since the density is constant, the velocity is given at boundary, and the pressure is totally determined by a Poisson equation. This is not the case for the compressible viscous flow simulation. The boundary variables are unknown before the interpolation. As a result, the weighted least square method cannot be used in compressible flow simulation directly. The boundary variables should be predetermined from the neighboring fluid

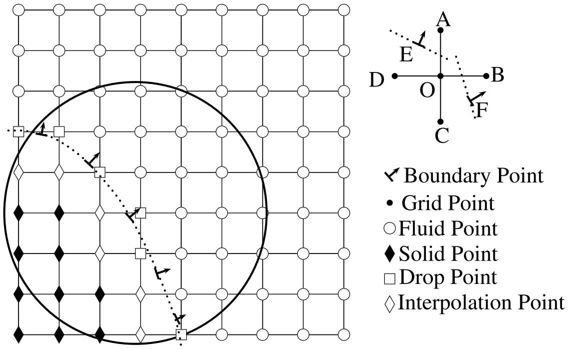

Fig. 2. The stencil for the boundary interpolation.

properties. For example, Palma et al. [16] assumed the constant pressure across the first layer of fluid point near the boundary, then use the derived boundary variables to construct the fluid variables at the interpolation point. In this study, we employ a constrained weighted least square method [23] to construct the fluid variables at the interpolation point. This interpolation process interpolates an intermediate flow state first, which is subsequently modified in order to satisfy the boundary condition. Then, the distribution of flow variables around the boundary is determined. Assume that the fluid variable around the boundary can be expressed as a polynomial,

$$P(x, y) = \sum_{p+q\leq m} c_{p,q} x^p y^q, \, p \geq 0, q \geq 0, \tag{19}$$

where m is the order of polynomial. Given the values $\{v_i\}$ are known at a set of locations $\{(x_i, y_i)|i = 1, \ldots, N, N \geq \dim(\mathbf{X})\}$, where N is the number of the fluid points in the boundary interpolation stencil (big circle in Fig. 2). And they should be approximately satisfied in the sense of minimizing a weighted mean square error function, which reads,

$$\text{minimize} \quad \sum_{i=1}^{N} (w_i(\mathbf{A}_i\mathbf{X} - v_i))^2, \tag{20}$$

where w_i is the weight which is written in an exponential form,

$$w_i = \exp(-\frac{\beta^2((x_i - x_b)^2 + (y_i - y_b)^2)}{(\max(\Delta x, \Delta y))^2}), \tag{21}$$

where β is an adjustable parameter. And v_i denotes the every single variable of $\mathbf{W}_i$.

$$\mathbf{X} = (c_{0,0}, \ \ldots, \ c_{m,0}, \ c_{0,1}, \ \ldots, \ c_{m-1,1}, \ \ldots, \ c_{0,m})^T, \tag{22}$$

$$\mathbf{A}_i = (1, \ \ldots, \ x_i^m y_i^0, \ x_i^0 y_i^1, \ \ldots, \ x_i^{m-1} y_i^1, \ x_i^0 y_i^m). \tag{23}$$

At first, the unconstrained problem described above is solved. Let $\mathbf{v} = (v_1, v_2, \ldots, v_N)$,

$$\mathbf{w} = \begin{pmatrix} w_0 & 0 & \ldots & 0 \\ 0 & w_1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & w_N \end{pmatrix}, \text{ and } \mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \ldots \\ \mathbf{A}_N \end{pmatrix}.$$

Then the solution is expressed as follows,

$$\mathbf{X} = \mathbf{C}^{-1}\mathbf{A}^T\mathbf{w}^2\mathbf{v}, \tag{24}$$

where $\mathbf{C} = \mathbf{A}^T\mathbf{w}^2\mathbf{A}$. And the value at boundary point (x_b, y_b) is evaluated by the value of the unconstrained polynomial $P^*(x_b, y_b)$.

Using this interpolation process, the intermediate boundary variable $\mathbf{W}_b^*$ is calculated. Then, to determine the pressure on the boundary, a second intermediate boundary density and pressure are calculated by

$$\left.\begin{aligned} \rho_b^{**} &= \rho_b^*(1 + \tfrac{1}{K+2}\mathrm{Ma}_{\mathbf{n}}^2)^{\frac{K+2}{2}} \\ p_b^{**} &= p_b^*(1 + \tfrac{1}{K+2}\mathrm{Ma}_{\mathbf{n}}^2)^{\frac{K+4}{2}} \end{aligned}\right\} \quad \text{for leeward surface where } (\mathbf{U}_b^* - \mathbf{U}_{wall}) \cdot \mathbf{n} > 0, \tag{25}$$

$$\left.\begin{aligned} \rho_b^{**} &= \rho_b^* \\ p_b^{**} &= p_b^* \end{aligned}\right\} \quad \text{for the other situation,} \tag{26}$$

where $\mathbf{n}$ denotes the normal direction of a boundary surface, and $\mathrm{Ma}_{\mathbf{n}}$ denotes the Mach number defined by the ratio of the orthogonal relative velocity to the boundary and the local speed of sound. The density and pressure assigned at the leeward surface are extremely important for supersonic flow, since the extrapolation from the downstream to the leeward surface is physically invalid at supersonic speed. Therefore, the stagnation point condition for one dimensional isentropic flow is adopted to construct the upstream flow condition at the leeward surface. Obviously, the intermediate boundary velocity is not the no-slip boundary condition or no-penetration boundary condition. Then, $\mathbf{W}_b^{**}$ is transformed to $\mathbf{W}_b$ to satisfy the boundary condition. To implement the isothermal no-slip boundary condition, we also assume that the pressure derived from $\mathbf{W}_b$ is identical to the pressure derived from $\mathbf{W}_b^{**}$. And the velocity and the temperature are replaced by the boundary velocity and temperature respectively. That is,

$$p_b = p_b^{**}, \tag{27}$$

$$\mathbf{U}_b = \mathbf{U}_{wall}, \tag{28}$$

$$\rho_b = p_b^{**}/(RT_{wall}), \tag{29}$$

$$\rho_b E_b = \frac{1}{2}\rho_b \mathbf{U}_b^2 + \frac{K+2}{2} p_b^{**}. \tag{30}$$

The Euler slip boundary condition (reflection boundary condition) presents,

$$p_b = p_b^{**}, \tag{31}$$

$$\mathbf{U}_b = (\mathbf{n}\cdot\mathbf{U}_{wall})\mathbf{n} + \mathbf{U}_b^{*} - (\mathbf{n}\cdot\mathbf{U}_b^{*})\mathbf{n}, \tag{32}$$

$$\rho_b = \rho_b^{**}, \tag{33}$$

$$\rho_b E_b = \frac{1}{2}\rho_b \mathbf{U}_b^2 + \frac{K+2}{2} p_b^{**}. \tag{34}$$

The no-slip adiabatic boundary condition gives,

$$p_b = p_b^{**}, \tag{35}$$

$$\mathbf{U}_b = \mathbf{U}_{wall}, \tag{36}$$

$$\rho_b = \rho^{**}, \tag{37}$$

$$\rho_b E_b = \frac{1}{2}\rho_b \mathbf{U}_b^2 + \frac{K+2}{2} p_b^{**}. \tag{38}$$

Then, the boundary variables $\mathbf{W}_b$ at several boundary points $\{(x_{b,j}, y_{b,j}) | j = 1, \dots, M, M < \dim(\mathbf{X})\}$ must satisfy the constraint,

$$\text{Constraint:} \quad \mathbf{A}_b\mathbf{X} = \mathbf{v}_b, \tag{39}$$

where $\mathbf{v}_b = (v_{b,1}, v_{b,2}, \dots, v_{b,M})$, $v_{b,j}$ denotes the every single component of $\mathbf{W}_{b,j}$, and M is the number of the boundary points which are chosen to be the constraint.

$$\mathbf{A}_b = \begin{pmatrix} 1 & \dots & x_{b,1}^m y_{b,1}^0 & x_{b,1}^0 y_{b,1}^1 & \dots & x_{b,1}^{m-1} y_{b,1}^1 & \dots & x_{b,1}^0 y_{b,1}^m \\ 1 & \dots & x_{b,2}^m y_{b,2}^0 & x_{b,2}^0 y_{b,2}^1 & \dots & x_{b,2}^{m-1} y_{b,2}^1 & \dots & x_{b,2}^0 y_{b,2}^m \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \dots & x_{b,M}^m y_{b,M}^0 & x_{b,M}^0 y_{b,M}^1 & \dots & x_{b,M}^{m-1} y_{b,M}^1 & \dots & x_{b,M}^0 y_{b,M}^m \end{pmatrix}. \tag{40}$$

Combining with Eq. (20), the constrained solution is

$$\mathbf{X} = \mathbf{C}^{-1}(I - \mathbf{A}_b^T \mathbf{C}_b^{-1}\mathbf{A}_b\mathbf{C}^{-1})\mathbf{A}^T\mathbf{w}^2\mathbf{v} + \mathbf{C}^{-1}\mathbf{A}_b^T\mathbf{C}_b^{-1}\mathbf{v}_b, \tag{41}$$

$$\text{with} \quad \mathbf{C}_b = \mathbf{A}_b\mathbf{C}^{-1}\mathbf{A}_b^T. \tag{42}$$

So we fully determine the polynomial $P(x, y)$ at the fluid side. If consider only one boundary point, then the current interpolation reproduces the least square method employed by Seo [21]. Furthermore, the unconstrained solution can be reused for calculating the constrained solution (for instance, $\mathbf{C}^{-1}$ is common part). In this study we only consider the linear interpolation. According to different boundary conditions, the derivatives at the interpolation points (or ghost points) on the solid side are determined by applying symmetric boundary condition. For the isothermal no-slip boundary,

$$\frac{\partial \rho}{\partial \mathbf{n}}|_{\text{fluid}} = \frac{\partial \rho}{\partial \mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial \rho}{\partial \mathbf{t}}|_{\text{fluid}} = \frac{\partial \rho}{\partial \mathbf{t}}|_{\text{solid}}, \tag{43}$$

$$\frac{\partial \mathbf{U_n}}{\partial \mathbf{n}}|_{\text{fluid}} = \frac{\partial \mathbf{U_n}}{\partial \mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial \mathbf{U_n}}{\partial \mathbf{t}}|_{\text{fluid}} = \frac{\partial \mathbf{U_n}}{\partial \mathbf{t}}|_{\text{solid}}, \tag{44}$$

$$\frac{\partial \mathbf{U_t}}{\partial \mathbf{n}}|_{\text{fluid}} = \frac{\partial \mathbf{U_t}}{\partial \mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial \mathbf{U_t}}{\partial \mathbf{t}}|_{\text{fluid}} = \frac{\partial \mathbf{U_t}}{\partial \mathbf{t}}|_{\text{solid}}, \tag{45}$$

$$\frac{\partial (\rho E)}{\partial \mathbf{n}}|_{\text{fluid}} = \frac{\partial (\rho E)}{\partial \mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial (\rho E)}{\partial \mathbf{t}}|_{\text{fluid}} = \frac{\partial (\rho E)}{\partial \mathbf{t}}|_{\text{solid}}; \tag{46}$$

for the Euler slip boundary,

$$\frac{\partial\rho}{\partial\mathbf{n}}|_{\text{fluid}} = -\frac{\partial\rho}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\rho}{\partial\mathbf{t}}|_{\text{fluid}} = \frac{\partial\rho}{\partial\mathbf{t}}|_{\text{solid}}, \tag{47}$$

$$\frac{\partial\mathbf{U_n}}{\partial\mathbf{n}}|_{\text{fluid}} = \frac{\partial\mathbf{U_n}}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\mathbf{U_n}}{\partial\mathbf{t}}|_{\text{fluid}} = -\frac{\partial\mathbf{U_n}}{\partial\mathbf{t}}|_{\text{solid}}, \tag{48}$$

$$\frac{\partial\mathbf{U_t}}{\partial\mathbf{n}}|_{\text{fluid}} = -\frac{\partial\mathbf{U_t}}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\mathbf{U_t}}{\partial\mathbf{t}}|_{\text{fluid}} = \frac{\partial\mathbf{U_t}}{\partial\mathbf{t}}|_{\text{solid}}, \tag{49}$$

$$\frac{\partial(\rho E)}{\partial\mathbf{n}}|_{\text{fluid}} = -\frac{\partial(\rho E)}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial(\rho E)}{\partial\mathbf{t}}|_{\text{fluid}} = \frac{\partial(\rho E)}{\partial\mathbf{t}}|_{\text{solid}}; \tag{50}$$

for the adiabatic no-slip boundary,

$$\frac{\partial\rho}{\partial\mathbf{n}}|_{\text{fluid}} = -\frac{\partial\rho}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\rho}{\partial\mathbf{t}}|_{\text{fluid}} = -\frac{\partial\rho}{\partial\mathbf{t}}|_{\text{solid}}, \tag{51}$$

$$\frac{\partial\mathbf{U_n}}{\partial\mathbf{n}}|_{\text{fluid}} = \frac{\partial\mathbf{U_n}}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\mathbf{U_n}}{\partial\mathbf{t}}|_{\text{fluid}} = \frac{\partial\mathbf{U_n}}{\partial\mathbf{t}}|_{\text{solid}}, \tag{52}$$

$$\frac{\partial\mathbf{U_t}}{\partial\mathbf{n}}|_{\text{fluid}} = \frac{\partial\mathbf{U_t}}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial\mathbf{U_t}}{\partial\mathbf{t}}|_{\text{fluid}} = \frac{\partial\mathbf{U_t}}{\partial\mathbf{t}}|_{\text{solid}}, \tag{53}$$

$$\frac{\partial(\rho E)}{\partial\mathbf{n}}|_{\text{fluid}} = -\frac{\partial(\rho E)}{\partial\mathbf{n}}|_{\text{solid}}, \qquad \frac{\partial(\rho E)}{\partial\mathbf{t}}|_{\text{fluid}} = -\frac{\partial(\rho E)}{\partial\mathbf{t}}|_{\text{solid}}, \tag{54}$$

where $\frac{\partial}{\partial\mathbf{n}}$ and $\frac{\partial}{\partial\mathbf{t}}$ denote the partial derivatives along the normal direction and the tangential direction of boundary surface respectively. Up to now, the linear interpolation polynomial at the solid side is totally determined, so is the variable at the interpolation point. This interpolation procedure is adopted for both low speed and supersonic flow simulations.

4. Numerical results

This section includes numerical examples from the subsonic incompressible limit to the hypersonic compressible viscous flow computations.

4.1. Subsonic flows

The isothermal no-slip boundary condition is adopted for all the subsonic flow simulations.

4.1.1. Lid-driven cavity flow

The lid-driven cavity flow is a classical benchmark problem. With the Cartesian grid method, it is not easy to simulate internal flows, since the current method theoretically cannot fully guarantee the mass conservation in the computational domain [17]. Owing to the mismatch between the flux point and the boundary point, the no penetrating condition can be satisfied in the sense of interpolation precision. Even though, it is still valuable to try the cavity case to estimate the conservative property of the Cartesian grid method.

The computational region is defined on a domain of $[-0.01, 1.01] \times [-0.01, 1.01]$, where two vertical walls locate at $x = 0$ and $x = 1$, two horizontal walls locate at $y = 0$ and $y = 1$. The length of the cavity is $L_0 = 1$. The upper wall is moving from left to right with a constant velocity. The isothermal boundary condition is adopted. The boundary values are given as follows,

$$\mathbf{U} = (U_0, 0), \quad T = 1, \quad 0 < x < 1, y = 1, \tag{55}$$

$$\mathbf{U} = (0, 0), \quad T = 1, \quad \text{the other boundaries.} \tag{56}$$

The number of grid points in each direction is 261. Since a margin adjacent to the solid wall is preserved, the internal area of cavity is approximately discretized by a 256 × 256 mesh (see the left bottom of the cavity in Fig. 3). The density is 1, and the temperature is 1 at the beginning. The right figure in Fig. 3 shows the total mass changes inside the cavity. The total mass should be a constant for a steady flow. However, the constant slope of the profile indicates that the total mass increases with time increasing. The mass changing rate is about 0.1% per unit time for this test case. The numerical results are compared with the Ghia's reference value. The velocity profiles at time $t = 30L_0/U_0$, when the total mass is not changing too much, are plotted in Fig. 4.

Although the boundary condition is not fully conservative, the velocity profiles are still quit good. The viscous effect can be simulated correctly in this case.

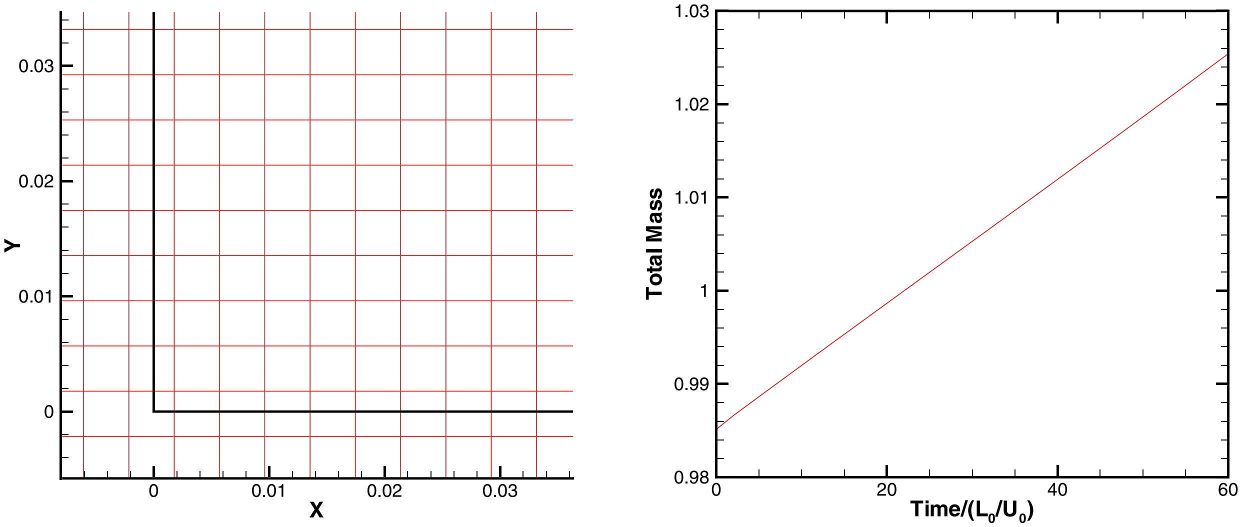

Fig. 3. The mesh at left bottom corner (left figure). The time series of the total mass of all the fluid points for the lid-driven cavity flow at Re = 1000 (right figure).

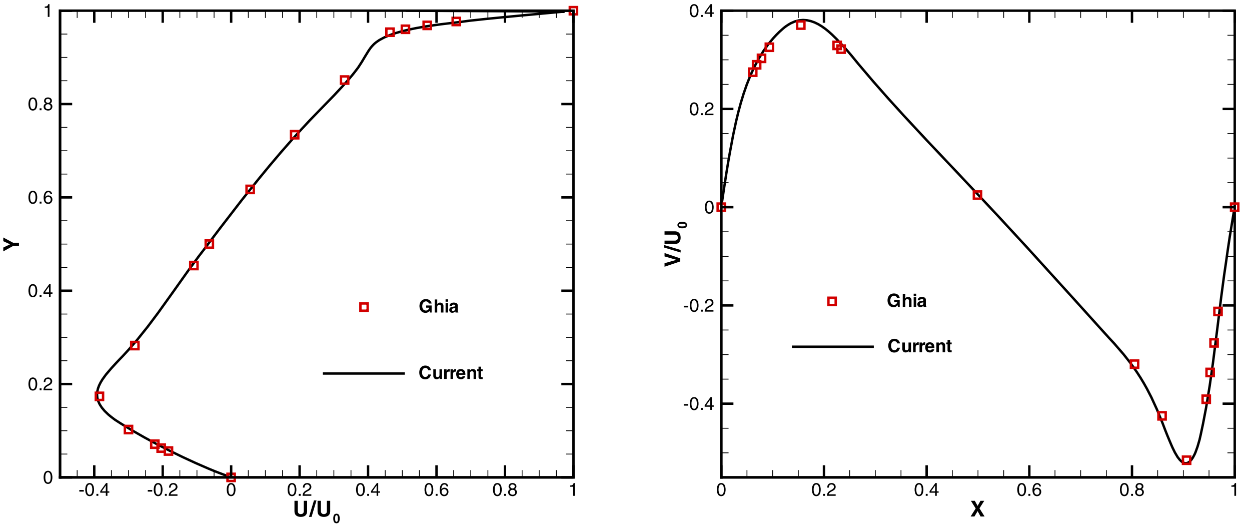

Fig. 4. The velocity U profile along the vertical central line and the velocity V profile along the horizontal central line for the lid-driven cavity flow at Re = 1000. The grid is 261 × 261 covering the range of $[-0.01, 1.01] \times [-0.01, 1.01]$.

4.1.2. Flow past a circular cylinder

The two dimensional flow passing through a circular cylinder is considered to test Cartesian grid method for steady flows at low Mach numbers. The free stream Mach is about 0.08, and Reynolds numbers based on the cylinder diameter, D, is Re = 10, 20, 30. A rectangular computational domain is used with the inlet and outlet vertical boundaries at $x = -15D$ and $x = 15D$, and the two horizontal far field boundaries at $y = \pm 15D$, respectively. Computations are performed using a grid of 903 × 903 points. The grid size is about $0.033D$. The curved boundaries are cut by the grid lines as shown in Fig. 5 on the left. The figure on the right shows the wake length at the rear of the cylinder. The pressure coefficient is defined by $C_p = (p - p_0)/(1/2(\rho_0 U_0^2))$, where the subscript '0' denotes the upstream or inlet flow condition. The reference data is extracted from the references [36–38]. Fig. 6 shows the pressure coefficient along the cylinder surface for Re = 20. The horizontal axis represents the angle of circumference where $\theta = 0$ and $\theta = 180$ correspond to the stagnation point and the rear of cylinder respectively. The agreement is quite satisfactory. It should be emphasized that the flow quantities at solid boundary are quit smooth in current computation owing to the constrained weighted least square interpolation.

4.1.3. Boundary accuracy

The low Reynolds number flow passing through a circular cylinder is planar symmetric in previous example. Therefore, the flow quantities should be exactly identical between the upper and lower half cylinder surfaces. However, if the cylinder

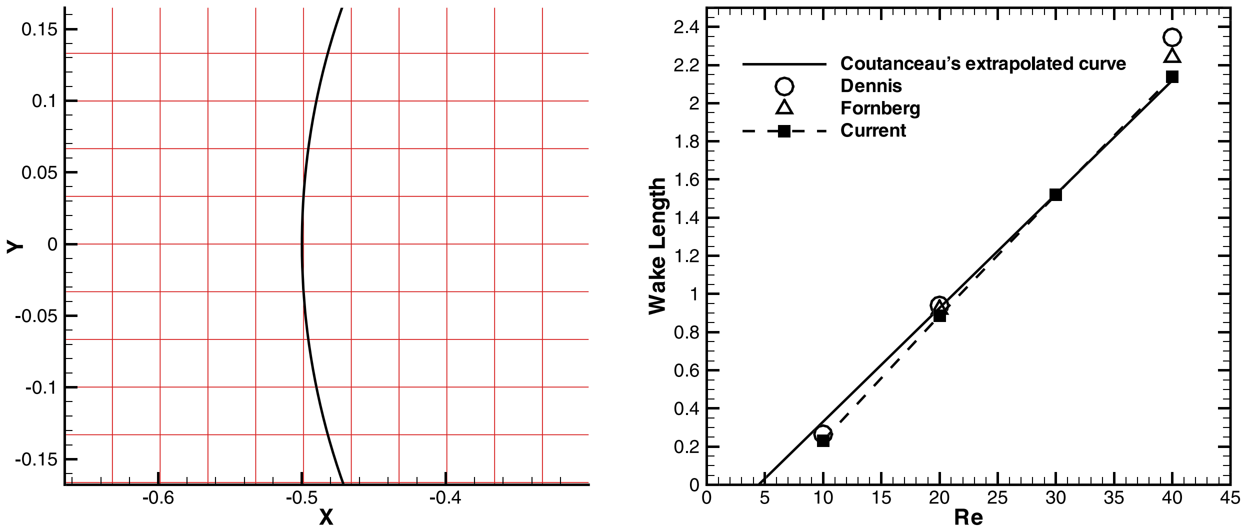

Fig. 5. The mesh around the cylinder (left figure). The wake length vs. Reynolds number for the flow past a circular cylinder (right figure).

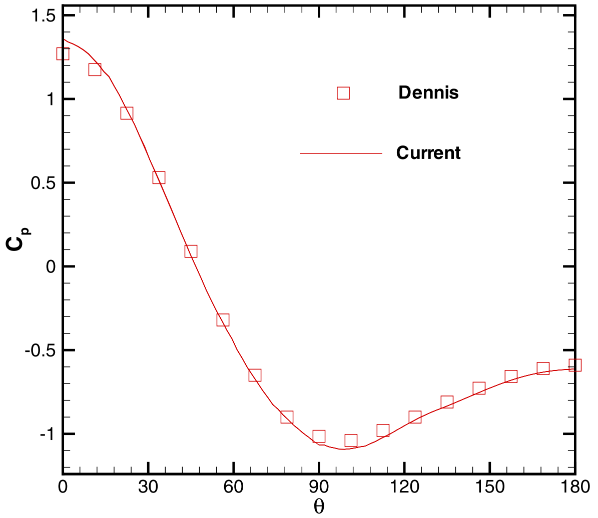

Fig. 6. The pressure coefficient along the circular cylinder surface for Re = 20.

is not placed symmetrically to a certain grid line along the x-axis, the geometrical locations of the upper and lower half cylinder surfaces will not be symmetric on discrete level. The error introduced from this kind of discrepancy can be considered as a measure of the accuracy of the Cartesian grid method. The L_1 error is defined as $Err_1 = \frac{1}{\pi}\int_0^{\pi} |p(\theta) - p(2\pi - \theta)| d\theta$; the L_∞ error is defined as $Err_\infty = \max_\theta\{p(\theta) - p(2\pi - \theta)\}$. We calculate the same problem on three successive refined meshes, i.e., 301×301, 602×602, 1204×1204. The computational domain covers the area, $[-5.004, 4.996] \times [-5.01, 4.99]$, and the center of the cylinder is $(0, 0)$. Obviously, the origin and the center of cylinder are mismatched. The flow condition is the same as the Reynolds 20 case in the last example. The errors are shown on the Table 1, and also plotted in Fig. 7. The current Cartesian grid method achieves an overall second order accuracy, and has only first order accuracy at some irregular points. This example totally excludes the influence of the flow solver. Thus, it is a good example for the accuracy evaluation for any Cartesian grid method.

4.1.4. Flow over a square cylinder

The 2D laminar flow around a square cylinder with length D mounted at the center of the channel is investigated. The configuration is the same as in the reference [39]. The blockage ratio is fixed at B = 1/8. So the height of channel is 8D. In order to reduce the influence of inflow and outflow boundary conditions, the length of the channel is set to be L/D = 35. The square cylinder locates at (0,0) where is 10D from the inflow boundary. The grid is 1416×341 points. The incoming flow's velocity profile is given as a parabolic curve. The maximum velocity in the middle of channel is 0.1. The initial gas density is uniformly 1. The inlet pressure is 1, and the pressure of outlet is given based on a typical pressure drop for

Table 1
The pressure discrepancy between the upper and lower surfaces of the cylinder for the symmetric flow on the asymmetric distributed meshes at Re = 20.

grid size (dx)	L_1 error	L_∞ error
0.03333	6.781×10^{-5}	2.140×10^{-4}
0.01664	2.346×10^{-5}	7.302×10^{-5}
0.00831	8.115×10^{-6}	4.697×10^{-5}

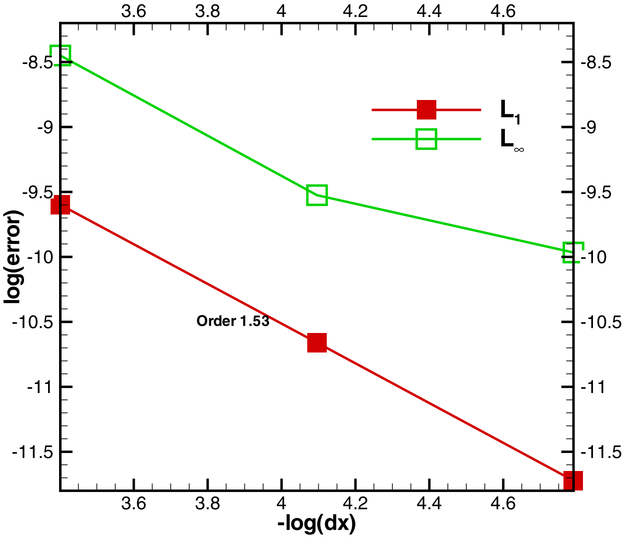

Fig. 7. The pressure discrepancy between the upper and lower surfaces of the cylinder vs. grid size on log scale.

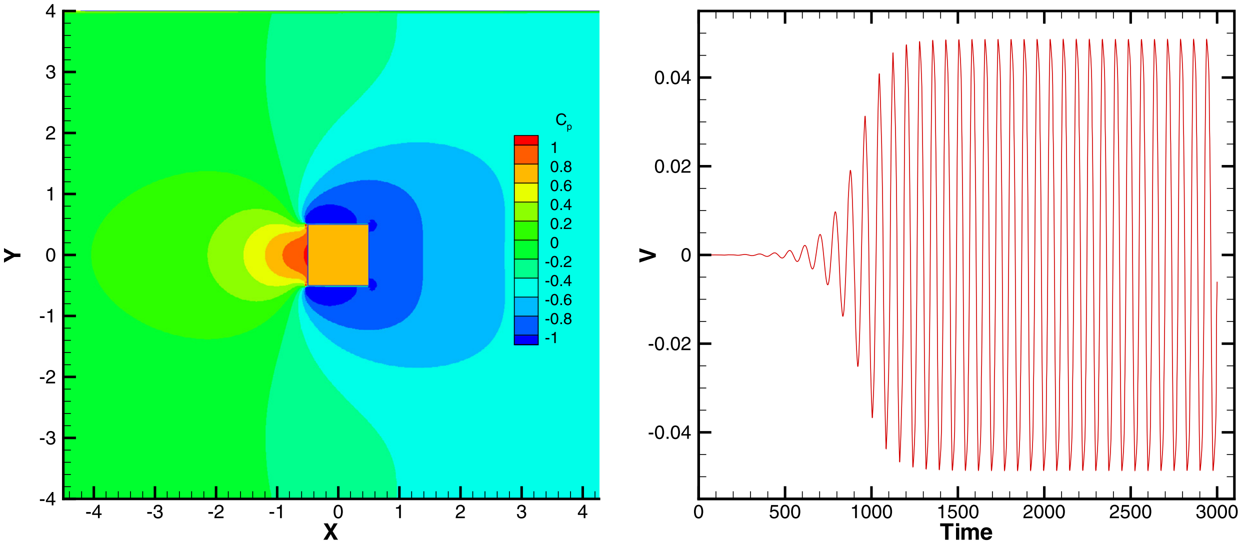

Fig. 8. The left figure is the pressure coefficient contour when Re = 20. The right figure is the vertical velocity series at the location (3.0, 0.0) behind the square when Re = 100. The corresponding Strouhal number is 0.137. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tube flow. Fig. 8 shows the pressure coefficient contour at Re = 20 and the time series of the unsteady vertical velocity at location (3.0, 0.0) when Re = 100. The Strouhal number derived from current method is about 0.137, which is identical to the reference data [39]. The wake length behind square is compared with reference data in Fig. 9. It also coincides with the numerical simulation by Breuer [39].

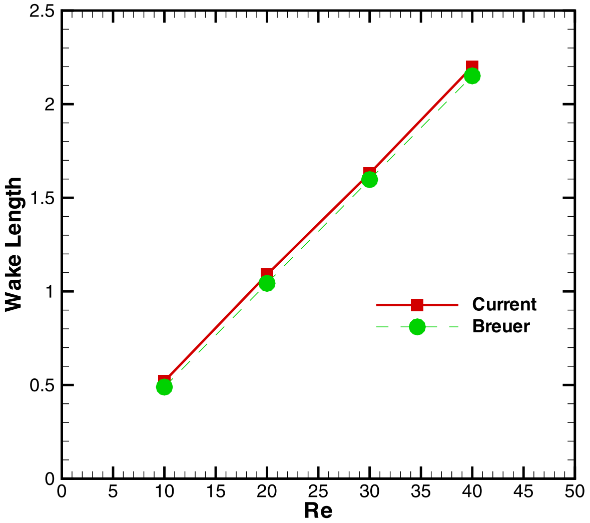

Fig. 9. The wake length vs. Reynolds number for the flow past a square cylinder.

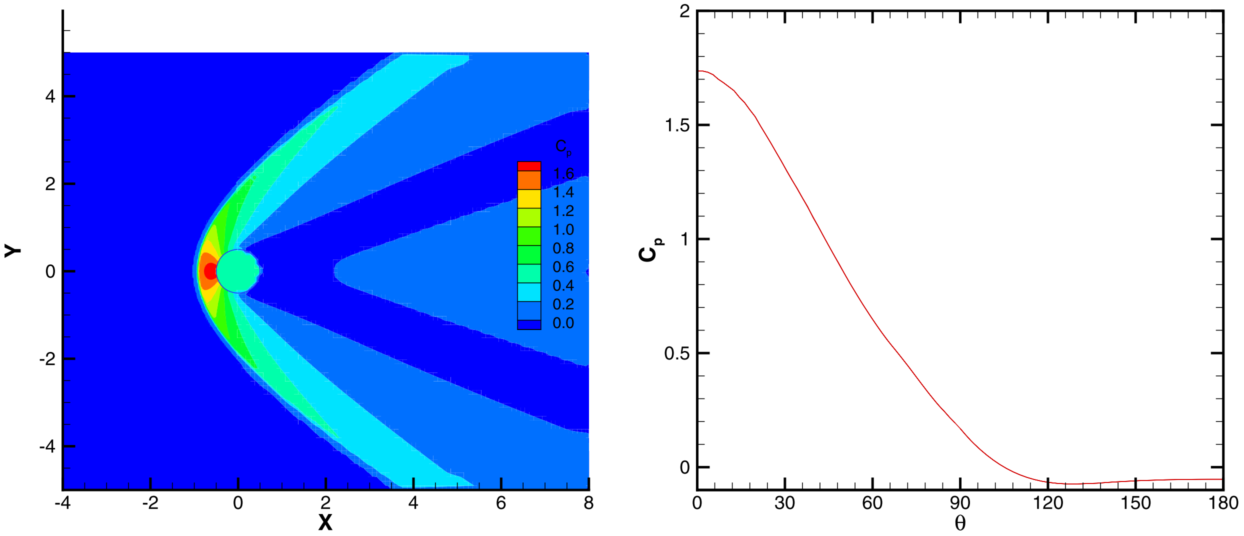

Fig. 10. The contour of the pressure coefficient (left) and the pressure coefficient (right) along the cylinder surface for the flow around a circular cylinder at Mach = 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.2. Supersonic flows

4.2.1. Supersonic flow around a circular cylinder

Flow around a blunt body is a very important test case for supersonic flow solver. We use this case to check the robustness of the current Cartesian grid method. The gas initially has a velocity of Mach 3. The Reynolds number defined by upstream condition is 1420. And the upstream pressure is 1. The grid with 363 × 303 mesh points covers the range of $[-4.0, 8.0] \times [-5.0, 5.0]$. Isothermal no-slip boundary condition is employed on the cylinder surface ($T_{wall} = 1$). The scheme survives initially at the strong rarefaction wave at the rear part of the cylinder. Fig. 10 shows the pressure coefficient contour and the pressure coefficient along the cylinder surface at the steady state. Then we put another two different geometries, say, a plate and a triangle, into the computational field. The flow condition is identical to the previous simulation. Fig. 11 shows the pressure coefficient contour and the stream lines. This simulation demonstrates the robustness and flexibility of current Cartesian grid method.

4.2.2. Mach 3 forward step flow

This two dimensional test problem has been studied by Chiang et al. [40] using Cartesian grid method. However, in their figures, the grid lines are superposed on the boundary. In our simulation, the computational domain is $[0.0, 3.0] \times [-0.05, 1.05]$. Two horizontal boundaries are placed at $y = 0.0$ and $y = 1.0$ respectively. A step with height 0.2 is located at

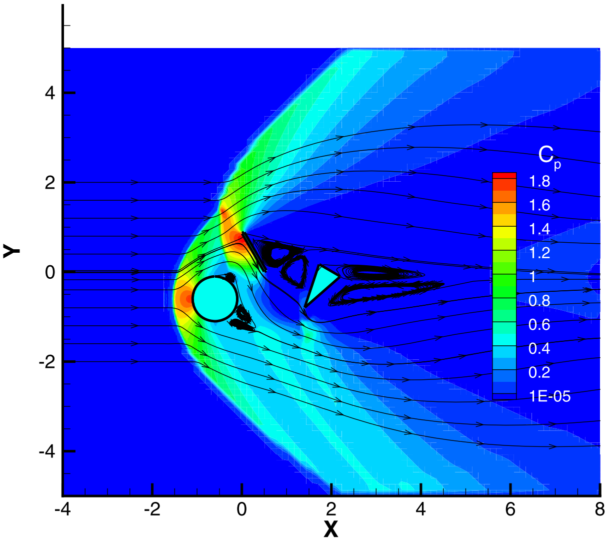

Fig. 11. The pressure coefficient contour and the stream lines for the flow past multiple objects when Ma = 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

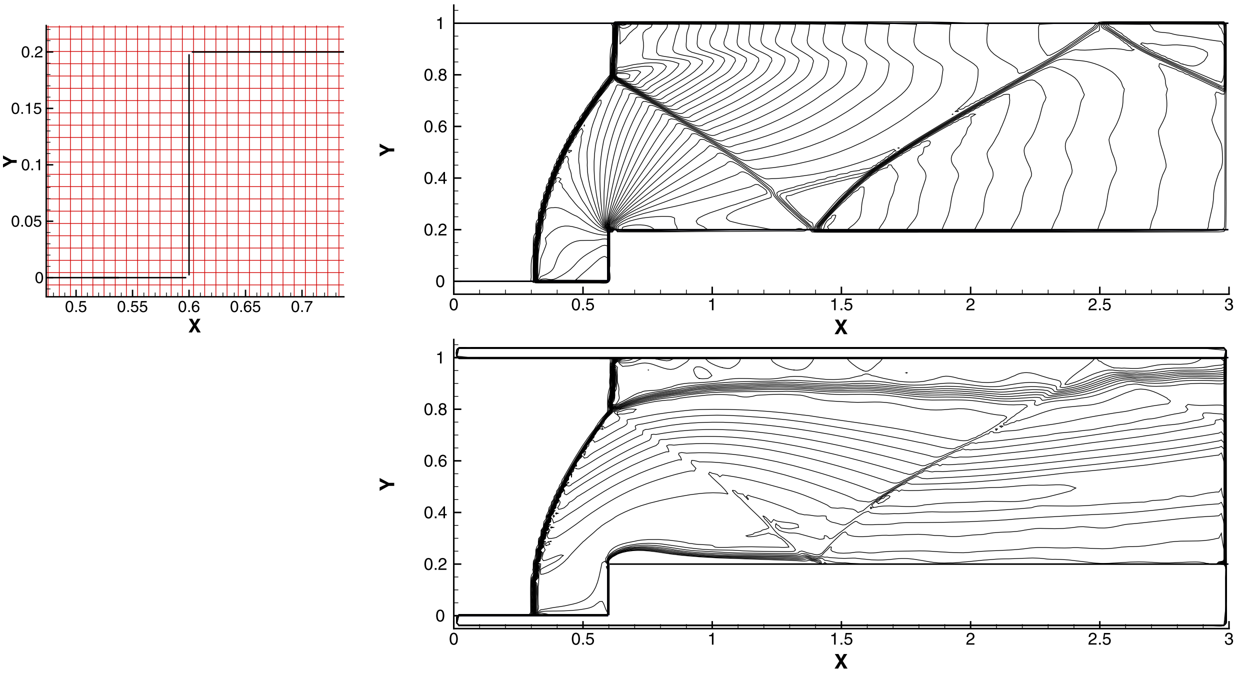

Fig. 12. The left figure illustrates the mesh around the step. The right top figure shows the density contour for Mach 3 step flow. The right bottom figure is the contour of p/ρ^γ. The grid is 304 × 102 covering the range of $[0.0, 3] \times [-0.05, 1.05]$.

$x = 0.6$. The upstream velocity is $(U, V) = (3, 0)$. The gas in the computational domain is uniform at the beginning, whose density is 1.4, pressure is 1.0, and velocity is 3. The Euler slip boundary condition is imposed at all solid boundaries. The horizontal direction is discretized by 304 points. Then we change the number of the vertical discrete points. The length of Mach stem at the top surface varies when the vertical discrete point number changes from 100 to 110. This is because that the minimum distance to the top boundary changes dramatically when the vertical discrete point number changes. And if double the mesh points in each direction, the numerical solution becomes insensitive to the small changes of the vertical discrete point number. Finally, the solution indeed converges to the reference solution [41] when the mesh is fine enough even if the shock exists near the solid boundary. The density distribution in the wind tunnel and the entropy (p/ρ^γ) at time $t = 4$ are shown in Fig. 12. No special treatment is adopted at the corner of the step.

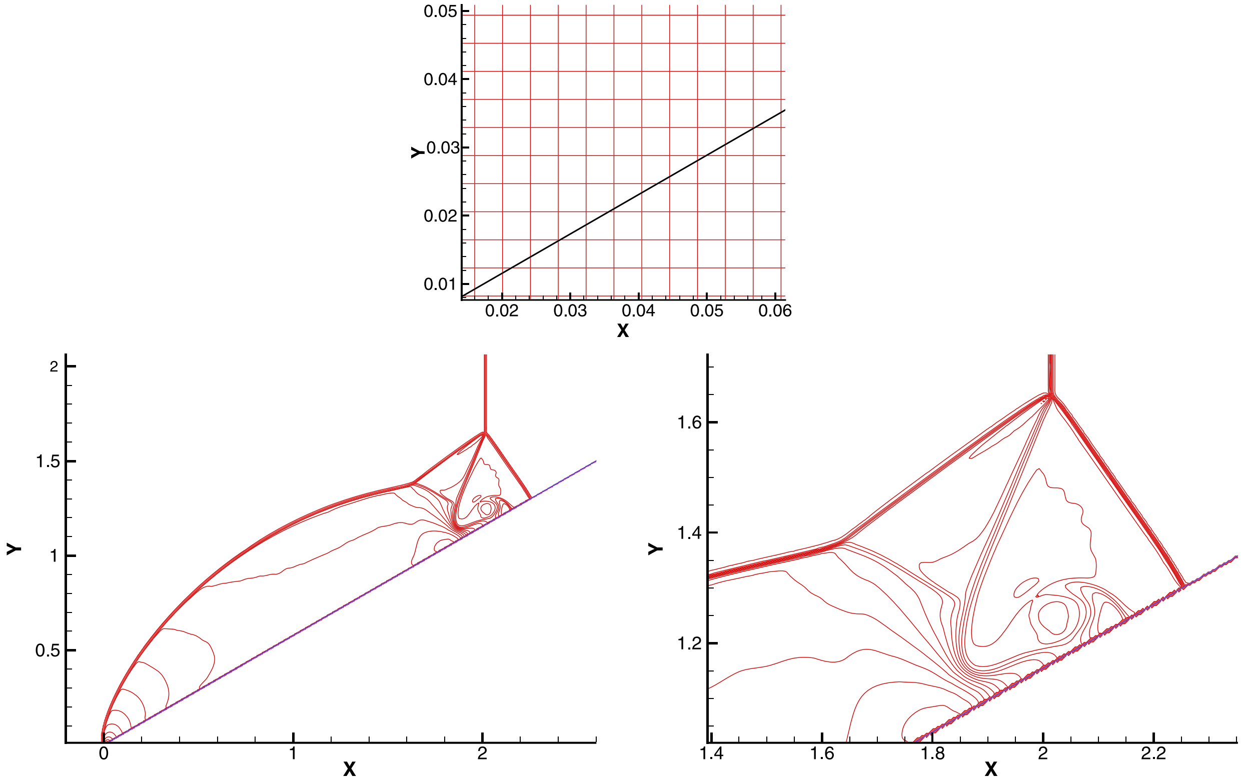

Fig. 13. The mesh is 900 × 730 covering the range of $[0.0, 3] \times [-0.2, 3.464]$. The top figure shows a portion of the mesh. The bottom figures show the density contour of double Mach reflection.

4.2.3. Double Mach reflection

A wedge is placed at the origin with 30 degree angle to x direction. The computational domain is $[-0.2, 3.464] \times [0.0, 3.0]$. The gas at $x > 0$, is at rest with density 1.4, and pressure 1. The flow condition at $x < 0$ is as follows, $\rho = 8$, $U = 8.25$, $p = 116.5$. A mesh of 900 × 730 points is used. The Euler slip boundary condition is imposed at the solid boundaries. Fig. 13 displays the density contours. This case has already been studied by Cartesian grid methods [17,12]. In Forrer's work [17], the whole structure moves slower than benchmark results. Our simulation results coincide with the results obtained by Pember [12] and the results from the calculation of a conformal mesh [42].

4.2.4. Viscous shock tube

This is a viscous problem introduced by Daru and Tenaud [43,44]. Sjögreen and Yee [45] and many other researchers also studied this problem. The gas with different thermal condition is initially at rest on both sides of $x = 0.5$, and separated by a membrane. Then, the membrane is removed and wave interaction occurs. The compressible Navier–Stokes equations with adiabatic no-slip boundary conditions are imposed. The solution develops complex two dimension shock/shear/boundary-layer interactions, which depend on the Reynolds number. The dimensionless initial states is given as follows,

$$\rho_l = 120,\ p_l = 120/\gamma,\ \rho_r = 1.2,\ p_r = 1.2/\gamma,$$

where subscripts "l" and "r" denotes the left side and right side of $x = 0.5$ respectively. The Prandtl number is 0.73. The case of Reynolds number 1000 till the time $t = 1.0$ is simulated. The computational domain is $[-0.51, 0.51] \times [-0.01, 0.51]$. A box of $[-0.5, 0.5] \times [0.0, 0.5]$ embraces the fluid domain. 2043 × 1023 grid points are used in this simulation. The bottom and two lateral boundaries are adiabatic with non-slip condition. The top boundary is reflection boundary (Euler slip boundary). Generally speaking, the structures of this problem (Fig. 14) are quit similar to that derived in the references [44,45]. This simulation demonstrates that current Cartesian grid method is capable of simulating complex supersonic viscous flow.

5. Conclusion

In this paper, we present a Cartesian grid method for complex immersed boundary problems, and use a simplified gas kinetic scheme for subsonic and supersonic flow simulation. A constrained weighted least square method is employed to update the physical quantities at the interpolation points. Different boundary conditions, including isothermal, adiabatic, and Euler slip conditions, are presented by different interpolation strategies. The new method is capable of simulating inviscid and viscous, compressible and near incompressible flow problems. The numerical results demonstrate that the new method

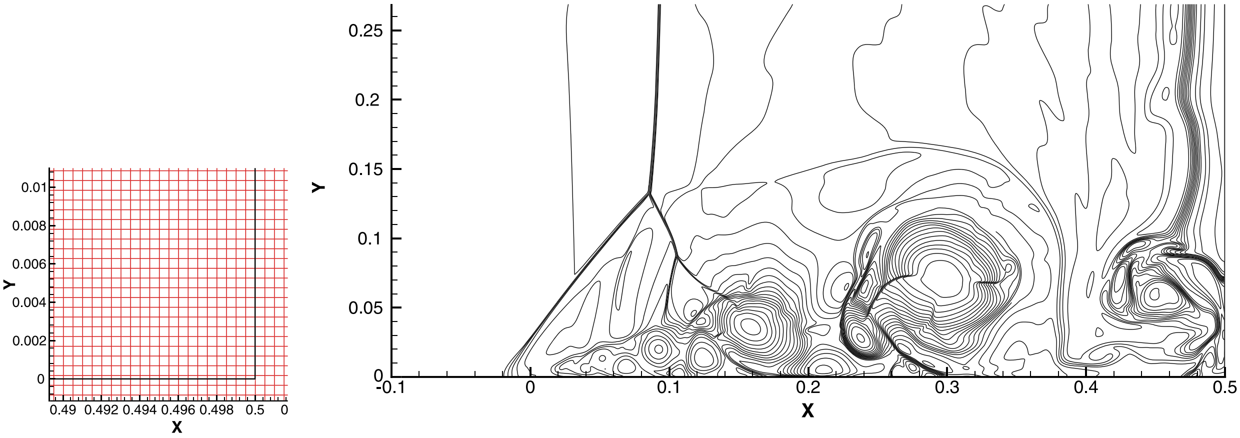

Fig. 14. The left figure shows the mesh around the right bottom corner of the computational domain. The right figure shows the density contour of viscous shock tube. The grid is 2043 × 1023 covering the range of $[-0.51, 0.51] \times [-0.01, 0.51]$.

converges to the correct physical solutions for both subsonic and supersonic flows as the mesh refines. The current scheme is robust under various flow condition from low Mach number to high Mach number flows, from inviscid to viscous ones. The interpolation procedure proposed in this study provides a smooth distribution of physical quantities at solid boundary and can tackle with arbitrary geometry. The current methodology can be further extended to the other flow systems, such as to the numerical schemes [46–48] for both continuum and rarefied flow computations.

Acknowledgements

This work was supported by Hong Kong Research Grant Council (16207715, 620813, 16211014), grants Provost13SC01 and SBI14SC11 at HKUST, National Nature Science Foundation of China under Grant Nos. 91530319, 11325212 and National Key Basic Research and Development Program (2014CB744100).

References

[1] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (2) (1999) 423–456.

[2] G. Caginalp, Stefan and Hele-Shaw type models as asymptotic limits of the phase-field equations, Phys. Rev. A 39 (11) (1989) 5887.

[3] H. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, J. Comput. Phys. 153 (2) (1999) 535–574.

[4] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, J. Comput. Phys. 135 (1) (1997) 8–29.

[5] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239–261.

[6] C.S. Peskin, The fluid dynamics of heart valves: experimental, theoretical, and computational methods, Annu. Rev. Fluid Mech. 14 (1) (1982) 235–259.

[7] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, J. Comput. Phys. 105 (2) (1993) 354–366.

[8] J. Mohd-Yusof, Combined Immersed-Boundary/B-Spline Methods for Simulations of Ow in Complex Geometries, Annual Research Briefs, NASA Ames Research Center, Stanford University Center of Turbulence Research, 1997, pp. 317–327.

[9] C. Shu, N. Liu, Y.-T. Chew, A novel immersed boundary velocity correction–lattice Boltzmann method and its application to simulate flow past a circular cylinder, J. Comput. Phys. 226 (2) (2007) 1607–1622.

[10] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (6) (2009) 1963–1979.

[11] M.J. Berger, R.J. Leveque, An adaptive cartesian mesh algorithm for the Euler equations in arbitrary geometries, AIAA paper 1930, 1989, AIAA-89-1930.

[12] R.B. Pember, J.B. Bell, P. Colella, W.Y. Curtchfield, M.L. Welcome, An adaptive cartesian grid method for unsteady compressible flow in irregular regions, J. Comput. Phys. 120 (2) (1995) 278–304.

[13] T. Ye, R. Mittal, H. Udaykumar, W. Shyy, An accurate cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (2) (1999) 209–240.

[14] H. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (1) (2001) 345–380.

[15] M.D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, J. Comput. Phys. 225 (2) (2007) 2098–2117.

[16] P. De Palma, M. De Tullio, G. Pascazio, M. Napolitano, An immersed-boundary method for compressible viscous flows, Comput. Fluids 35 (7) (2006) 693–702.

[17] H. Forrer, R. Jeltsch, A higher-order boundary treatment for cartesian-grid methods, J. Comput. Phys. 140 (2) (1998) 259–277.

[18] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, J. Comput. Phys. 225 (1) (2007) 528–553.

[19] F. Gibou, R.P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (1) (2002) 205–227.

[20] Y. Peng, L.-S. Luo, A comparative study of immersed-boundary and interpolated bounce-back methods in LBE, Prog. Comput. Fluid Dyn., Int. J. 8 (1) (2008) 156–167.

[21] J.H. Seo, R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced sound in complex geometries, J. Comput. Phys. 230 (4) (2011) 1000–1019.

[22] L. Duan, X. Wang, X. Zhong, A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness, J. Comput. Phys. 229 (19) (2010) 7207–7237.

[23] D.-W. Zhang, F.-S. Wei, Exposition and argumentation on constrained least square method, Chin. J. Comput. Mech. 17 (4) (2000) 398–404.

[24] K. Xu, L. Martinelli, A. Jameson, Gas-kinetic finite volume methods, flux-vector splitting, and artificial diffusion, J. Comput. Phys. 120 (1) (1995) 48–65.

[25] K. Xu, A. Jameson, Gas-kinetic relaxation (BGK-type) schemes for the compressible Euler equations, AIAA-95-1736-CP, in: 12th AIAA Computational Fluid Dynamics Conference, San Diego, CA, June 1995, pp. 19–22.

[26] K. Xu, Gas-kinetic schemes for unsteady compressible flow simulations, VKI for Fluid Dynamics Lecture Series.

[27] K. Xu, A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys. 171 (2001) 289–335.

[28] K. Xu, Direct Modeling for Computational Fluid Dynamics: Construction and Application of Unified Gas-Kinetic Schemes, World Scientific, Singapore, 2015.

[29] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases, Phys. Rev. 94 (1954) 511.

[30] Z.-H. Li, H.-X. Zhang, Study on gas kinetic unified algorithm for flows from rarefied transition to continuum, J. Comput. Phys. 193 (2004) 708–738.

[31] Z.-H. Li, H.-X. Zhang, Gas-kinetic numerical studies of three-dimensional complex flows on spacecraft re-entry, J. Comput. Phys. 228 (2009) 1116–1138.

[32] S. Chen, K. Xu, C. Lee, Q. Cai, A unified gas kinetic scheme with moving mesh and velocity space adaptation, J. Comput. Phys. 231 (20) (2012) 6643–6664.

[33] L. Zhu, Z. Guo, K. Xu, Discrete unified gas kinetic scheme on unstructured meshes, Comput. Fluids 127 (2016) 211–225.

[34] R. Yuan, C. Zhong, H. Zhang, An immersed-boundary method based on the gas kinetic BGK scheme for incompressible viscous flow, J. Comput. Phys. 296 (2015) 184–208.

[35] Y. Sun, C. Shu, Y. Wang, C. Teo, Z. Chen, An immersed boundary-gas kinetic flux solver for simulation of incompressible flows, Comput. Fluids (2016), http://dx.doi.org/10.1016/j.compfluid.2016.01.007, in press.

[36] M. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (02) (1977) 231–256.

[37] S. Dennis, G.-Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100, J. Fluid Mech. 42 (03) (1970) 471–489.

[38] B. Fornberg, A numerical study of steady viscous flow past a circular cylinder, J. Fluid Mech. 98 (04) (1980) 819–855.

[39] M. Breuer, J. Bernsdorf, T. Zeiser, F. Durst, Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume, Int. J. Heat Fluid Flow 21 (2) (2000) 186–196.

[40] Y.-L. Chiang, B. van Leer, K.G. Powell, Simulation of unsteady inviscid flow on an adaptively refined Cartesian grid, AIAA paper, 1992, AIAA-92-0443.

[41] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1) (1984) 115–173.

[42] J. Qiu, C.-W. Shu, Finite difference WENO schemes with Lax–Wendroff-type time discretizations, SIAM J. Sci. Comput. 24 (6) (2003) 2185–2198.

[43] V. Daru, C. Tenaud, Evaluation of TVD high resolution schemes for unsteady viscous shocked flows, Comput. Fluids 30 (1) (2001) 89–113.

[44] V. Daru, C. Tenaud, Numerical simulation of the viscous shock tube problem by using a high resolution monotonicity-preserving scheme, Comput. Fluids 38 (3) (2009) 664–676.

[45] B. Sjögreen, H. Yee, Grid convergence of high order methods for multiscale complex unsteady viscous compressible flows, J. Comput. Phys. 185 (1) (2003) 1–26.

[46] K. Xu, J. Huang, An improved unified gas-kinetic scheme and the study of shock structures, IMA J. Appl. Math. 76 (2011) 698–711.

[47] J. Huang, K. Xu, P. Yu, A unified gas-kinetic scheme for continuum and rarefied flows III: microflow simulations, Commun. Comput. Phys. 14 (2013) 1147–1173.

[48] S. Chen, K. Xu, A cartesian grid-based unified gas kinetic scheme, in: Proceedings of the 29th International Symposium on Rarefied Gas Dynamics, vol. 1628, AIP Publishing, 2014, pp. 995–1002.