Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 222 (2007) 155–175
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A unified moving grid gas-kinetic method in Eulerian space for viscous flow computation

Changqiu Jin, Kun Xu *

Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China

Received 6 March 2006; received in revised form 8 July 2006; accepted 19 July 2006
Available online 1 September 2006

Abstract

Under a generalized coordinate transformation with arbitrary grid velocity, the gas-kinetic BGK equation is reformulated in a moving frame of reference. Then, a unified conservative gas-kinetic scheme is developed for the viscous flow computation in the moving grid system in the Eulerian space. Due to the coupling between the grid velocity and the overall solution algorithm, the Eulerian and Lagrangian methods become two limiting cases in the current gas-kinetic method. A fully conservative formulation can be obtained even in the Lagrangian limit. The moving grid method extends the applicable regime of the gas-kinetic scheme to the flows with free surface or moving boundaries, such as dam break problem and airfoil oscillations. In order to further increase the robustness of the moving grid method, similar to the arbitrary Lagrangian–Eulerian (ALE) method, a conservative adaptive grid technique is also implemented in the current method to redistribute the mesh concentration to the rapid variational flow region and remedy the distorted moving mesh due to the coupling between grid velocity and fluid speed. Many numerical examples from incompressible flow to the supersonic shock interaction are presented. The test cases verify the accuracy and robustness of the unified moving grid gas-kinetic method.

MSC: 65M06; 76P05; 76T05

Keywords: Unified coordinate system; Moving grid; Gas-kinetic scheme; Navier–Stokes equations

1. Introduction

There are two different coordinate system for description of fluid motion: the Eulerian one describes fluid motion at fixed locations, and the Lagrangian one follows fluid particles. Considerable progress has been made over the past two decades on developing computational fluid dynamics (CFD) methods based on the above two coordinates system. As the unsteady flow calculations with moving boundaries and interfaces become important, such as found in the flutter simulation of wings, turbomachinery blades, and multiphase flow, the development of fast and reliable methods for dynamically deforming computational domain is

* Corresponding author. Tel.: +852 2358 7440; fax: +852 2358 1643.
E-mail addresses: majchq@ust.hk (C. Jin), makxu@ust.hk (K. Xu).

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.07.015

required [17]. This research will help significantly the industry, such as the aerodynamic shape optimization studies and the detonative chemical reactive flow computation.

There are many moving mesh methods in the literature. One example is the static mesh movement method, where the new mesh is generated at each time step according to certain monitor function and the flow variables are interpolated into the newly generated mesh. Then, the flow update through the cell interface fluxes is done on a static mesh. In order to increase the accuracy, the mesh can be properly adapted [9]. Another example is the dynamical one, where the mesh is moving according to certain velocity. At the same time, the fluid variables are updated inside each moving control volume within a time step. The second method is mostly used to track the interface location, to account for changes in the interface topology, and to resolve small-scale structure at singular point. The most famous one for this dynamical mesh moving method is the Lagrangian method. Through the research in the past decades, it has been well recognized that the Lagrangian method is always associated with the mesh tangling once the fluid velocity is used as the mesh moving velocity. In order to avoid severe mesh distortion in the Lagrangian method, many techniques have been developed. The widely used one at present time is the arbitrary Lagrangian–Eulerian (ALE) technique, which uses continuous re-zoning and re-mapping from Lagrangian to the Eulerian grid. Unfortunately, this process requires interpolations of geometry and flow variables [14]. In aerospace engineering, in order to re-distribute the boundary deformation dynamically into the whole computational domain a spring network approach has been usually used to determine the motion of the mesh point, such as those around a deforming airfoil [2,16,23]. Here, a smoothing global operator is applied in maintaining grid smoothness and grid angles. This process is always associated with iterative methods resembling an elliptic grid generator. With a general transformation between the physical (t, x, y) and the computational space (λ, ξ, η), the Navier–Stokes equations can be written in a conservative form [7]. Many numerical schemes have been developed based on the above formulation for the Navier–Stokes equations directly, such as in the cases of fluid-structure interaction and fluid induced vibration. Instead of constructing an exact Riemann solver, an efficient approximate Riemann solution has been obtained [6], where the grid velocity is explicitly used in the wave decomposition. Even without using conservative governing equations explicitly, many moving mesh methods for incompressible Navier–Stokes equations, hyperbolic system, or chemical reactive flow, have also been developed with detailed consideration of numerical cell deformation [20,4,1].

Recently, a successful moving mesh method for inviscid Euler equations has been developed by Hui et al. [10] on the target of crisp capturing of slip line. In this unified coordinate method, with a prescribed grid velocity, the inviscid flow equations are written in a conservative form in the computational domain (λ, ξ, η), as well as the geometric conservation laws which control the mesh deformation. The most distinguishable merit in the unified coordinate method is that the fluid equations and geometric evolution equations are written in a combined system, which is different from the fluid equations alone [7]. Furthermore, due to the coupling of the fluid and geometric system, for the first time the multi-dimensional Lagrangian gas dynamic equations have been written in a conservative form. As a consequence, theoretically it has been shown that the multi-dimensional Lagrangian system is only weakly hyperbolic. Numerically, in the unified coordinate system the fluid and geometric variables can be updated simultaneously. In order to overcome the disadvantage in the Lagrangian method, in the unified coordinate system the grid velocity is set to be $h\mathbf{q}$, where $\mathbf{q} = (U, V)$ is the fluid velocity and h is a parameter which is to be determined by conditions, such as the mesh alignment with the slip surface, or keeping grid angle during the mesh movement. Therefore, the grid velocity can be changed locally according to the value of h. In a recent paper [11], the grid velocity has been further generalized to (hU, kV), where h and k are two parameters to be determined. The great achievement of the unified coordinate method is that the numerical diffusion across the slip line is reduced to a minimum with the crisp capturing of contact discontinuity. However, in the complicated flow movement, in order to avoid the severe mesh distortion, the constraints, such as keeping mesh orthogonality and grid angles, have to be used in the unified coordinate system. As a result, in most cases, the constraint automatically enforces the mesh velocity being zero, such as in the case of gas implosion inside a square. Otherwise, for flow problems with circulations, any mesh movement method, once the grid speed is coupled with the fluid velocity, will distort the mesh eventually and stop the computation. Also, in order to capture the slip line, the unified method is mainly focusing on the solution of the Euler equations. For the viscous flow, the equations, see Appendix, become much more complicated in a unified coordinate system.

Based on the gas-kinetic Boltzmann equation, the Navier–Stokes equations can be derived using the Chapman–Enskog expansion [5]. In the gas-kinetic representation, all macroscopic flow variables are the moments of a single particle distribution function and the particle movement is basically the linear transport and collision. In the past years, we have been concentrated on developing the gas-kinetic scheme in the Eulerian space with stationary grid. The gas-kinetic BGK scheme has been well developed for the compressible Navier–Stokes solutions [24], and the scheme is especially accurate for the supersonic viscous and heat conducting flow [25,19]. Since a gas distribution function includes both equilibrium and non-equilibrium flow properties, the inviscid and viscous fluxes are obtained simultaneously in the gas-kinetic scheme.

In the gas-kinetic approach, the gas distribution function is a scalar function, which can be easily transferred from one reference of frame to another one once the relative speed between them is given. The purpose of this paper is to develop a unified gas-kinetic method based on the transformation in [10]. Firstly, the gas-kinetic BGK equation is transformed from the physical space (t, x, y) to the computational space (λ, ξ, η). Then, the corresponding conservative Euler and Navier–Stokes equations in the computational domain are obtained using the Chapman–Enskog expansion. In the computational space (λ, ξ, η), the gas-kinetic BGK equation is solved for the solution of the gas distribution function in the computational domain, which is subsequently used to evaluate the flux across a moving cell interface in the Eulerian space in order to update the flow variables inside each controlling volume with moving boundaries. Since the inviscid and viscous fluxes are included simultaneously in the gas-kinetic formulation, the Navier–Stokes fluxes are obtained automatically across the moving cell interface. The advantage of the current method, as for the unified coordinate system [10], is that an exact conservative mathematical formulation for both fluid and geometrical variables is used and kept in the numerical scheme. If the unified system is not used, the satisfaction of geometrical conservation laws are mostly based on the physical intuition in the multi-dimensional case, such as the moving mesh approach using flux vector splitting method for the Euler equations [15]. Since the gas-kinetic scheme solves the viscous governing equations intrinsically, different from Hui's unified method for the Euler equations it has no the ability to resolve the slip line discontinuity without any transition point. In other words, the dissipation in the kinetic scheme will take effect to smear the contact discontinuity, but the magnitude of smearing under the unified coordinate transformation will be much reduced in comparison with the scheme without mesh moving, especially for the cases that the physical thickness cannot be resolved by the cell size. Furthermore, in order to avoid the severe mesh distortion due to the coupling of grid speed with the fluid velocity, an adaptive mesh method is going to be used to regularize the mesh distribution in the domain. In other words, once the quality of the mesh is deteriorated, at a fixed time instant the grid points are redistributed according to a monitor function, and all corresponding conservative flow variables are interpolated from the old to a newly generated mesh [21,13]. The function of this step is two folds. Firstly, it is necessary for a unified scheme to have some mechanism to regularize the mesh from tangling in order to make the scheme robustness. Secondly, for the current viscous flow computation, by using a proper monitor function in the adaptive mesh step the grid points can be easily concentrated on the high gradient regions, where the accuracy of the solution can be much improved, such as inside the boundary layer. With the unique coupling between the moving and adaption in the unified scheme, we have successfully simulated the free surface flow using the Lagrangian gas-kinetic scheme.

This paper is organized as the following. Section 2 is about the mathematical formulation of the gas-kinetic BGK model in a unified coordinate system and the construction of the gas-kinetic scheme. Section 3 is about the numerical experiments, where many examples from free surface flow to the supersonic flow reflections are presented. The last section is the conclusion.

2. A gas-kinetic BGK scheme under unified coordinate system

2.1. Gas-kinetic BGK model under coordinate transformation

The BGK model of the approximate Boltzmann equation in two-dimensional space (2D) can be written as [3]:

$$f_t + uf_x + vf_y = \frac{g - f}{\tau}, \tag{1}$$

where f is the gas distribution function and g is the equilibrium state approached by f. Both f and g are functions of space (x, y), time t, particle velocity (u, v), and internal variable ζ. As presented in [10], a coordinate transformation can be used from the physical domain (t, x, y) to the computational domain (λ, ξ, η):

$$\begin{cases} \mathrm{d}t = \mathrm{d}\lambda, \\ \mathrm{d}x = U_g \mathrm{d}\lambda + A\mathrm{d}\xi + L\mathrm{d}\eta, \\ \mathrm{d}y = V_g \mathrm{d}\lambda + B\mathrm{d}\xi + M\mathrm{d}\eta, \end{cases} \tag{2}$$

where (U_g, V_g) are the grid velocity, and (A, B, L, M) are determined by the compatibility conditions or the geometric conservative laws:

$$\begin{cases} \frac{\partial A}{\partial \lambda} = \frac{\partial U_g}{\partial \xi}, \\ \frac{\partial B}{\partial \lambda} = \frac{\partial V_g}{\partial \xi}, \\ \frac{\partial L}{\partial \lambda} = \frac{\partial U_g}{\partial \eta}, \\ \frac{\partial M}{\partial \lambda} = \frac{\partial V_g}{\partial \eta}. \end{cases} \tag{3}$$

With the above transformation (2), the gas-kinetic BGK equation becomes:

$$\frac{\partial}{\partial \lambda}(\Delta f) + \frac{\partial}{\partial \xi}\{[(u - U_g)M - (v - V_g)L]f\} + \frac{\partial}{\partial \eta}\{[-(u - U_g)B + (v - V_g)A]f\} = \frac{g - f}{\tau}\Delta, \tag{4}$$

where $\Delta = AM - BL$ is the Jacobian of the transformation. The schematic relation between particle velocity, flow velocity, and mesh moving velocity is shown in Fig. 1.

For an equilibrium flow with distribution $f = g$, by taking the conservative moments $\phi = \left(1, u, v, \frac{1}{2}(u^2 + v^2 + \zeta^2)\right)^{\mathrm{T}}$ to Eq. (4), the corresponding Euler equations in computational domain in the Eulerian space can be obtained:

$$\frac{\partial}{\partial \lambda}\begin{pmatrix} \rho\Delta \\ \rho\Delta U \\ \rho\Delta V \\ \rho\Delta E \end{pmatrix} + \frac{\partial}{\partial \xi}\begin{pmatrix} \rho(I - I_g) \\ \rho U(I - I_g) + PM \\ \rho V(I - I_g) - PL \\ \rho E(I - I_g) + PI \end{pmatrix} + \frac{\partial}{\partial \eta}\begin{pmatrix} \rho(J - J_g) \\ \rho U(J - J_g) - PB \\ \rho V(J - J_g) + PA \\ \rho E(J - J_g) + PJ \end{pmatrix} = 0, \tag{5}$$

where U and V are fluid velocity in the x- and y-directions, $I = UM - VL$, $I_g = U_g M - V_g L$, $J = AV - BU$, and $J_g = AV_g - BU_g$. Note again that the time evolution of the physical quantities is still in the Eulerian space. The difference between the Euler equations in Eulerian space and the above system is due to the mesh moving in space and time.

For the viscous and heat conducting flow, the Chapman–Enskog expansion of Eq. (4) up to the first-order of τ gives:

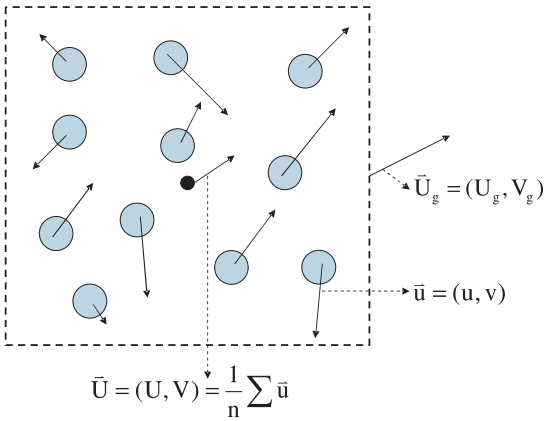


Fig. 1. Schematic relation between particle velocity (u, v), flow velocity (U, V), and mesh moving velocity (U_g, V_g).

$$f = g - \frac{\tau}{\Delta}\left(\frac{\partial}{\partial\lambda}(\Delta g) + \frac{\partial}{\partial\xi}\{[(u-U_g)M - (v-V_g)L]g\} + \frac{\partial}{\partial\eta}\{[-(u-U_g)B + (v-V_g)A]g\}\right).$$

Taking moments of ϕ again to Eq. (4) with the above NS distribution function, we can get the Navier–Stokes equations in moving space, which are presented in Appendix. Numerically, instead of solving the viscous governing equations in Appendix, we are going to solve the gas-kinetic equation for the viscous solution.

2.2. Numerical BGK–NS scheme in a moving mesh system

In this section, we are going to present the gas-kinetic method to solve Eq. (4) by a directional splitting method. For example, the BGK model (4) in the ξ-direction is:

$$\frac{\partial}{\partial\lambda}(\Delta f) + \frac{\partial}{\partial\xi}\{[(u-U_g)M - (v-V_g)L]f\} = \frac{g-f}{\tau}\Delta. \tag{6}$$

In order to evaluate the fluxes across a moving interface $\xi = \text{constant}$, let us first define its normal direction and tangent directions:

$$\vec{n} = \nabla\xi/|\nabla\xi| = (M,-L)/S, \quad \vec{t} = (L,M)/S,$$

where $S = \sqrt{M^2+L^2}$ is the physical length of the cell interface. Then, the particle velocity $(u-U_g,\ v-V_g)$ relative to a moving cell interface can be decomposed into the normal $\tilde{u}$ and tangential $\tilde{v}$ velocities as well, namely,

$$\begin{cases} \tilde{u} = (u-U_g)M/S - (v-V_g)L/S, \\ \tilde{v} = (u-U_g)L/S + (v-V_g)M/S. \end{cases} \tag{7}$$

Hence, with the above transformation, Eq. (6) in the ξ-direction becomes:

$$\frac{\partial}{\partial\lambda}(\Delta f) + \frac{\partial}{\partial\xi}(S\tilde{u}f) = \frac{g-f}{\tau}\Delta. \tag{8}$$

This is the basic equation to be solved to construct the gas distribution function f at the moving cell interface $\xi = \text{constant}$, then calculate the numerical fluxes. In the above equation, Δ is basically the cell area and S is the cell interface length. At the center of a moving cell interface the above equation can be re-written as

$$\frac{\partial}{\partial\lambda}(f) + \frac{\partial}{\partial\tilde{x}}(\tilde{u}f) = \frac{g-f}{\tau}, \tag{9}$$

where $\tilde{x}$ is the length scale in the normal direction of the moving cell interface in the physical space. Since $\mathrm{d}\lambda = \mathrm{d}t$, the integral solution of the above equation becomes

$$f(\xi_{i+1/2},\eta_j,t,\tilde{u},\tilde{v},\zeta) = \frac{1}{\tau}\int_0^t g(\tilde{x}',t',\tilde{u},\tilde{v},\zeta)\mathrm{e}^{-(t-t')/\tau}\mathrm{d}t' + \mathrm{e}^{-t/\tau}f_0(\tilde{x}_{i+1/2}-\tilde{u}t,\tilde{v}), \tag{10}$$

where $\tilde{x}' = \tilde{x}_{i+1/2} - \tilde{u}(t-t')$ is the trajectory of a particle motion relative to the moving cell interface and f_0 is the initial gas distribution function f at the beginning of each time step ($t = 0$). The scheme based on the above solution will be identical to the BGK–NS method [24], even though the system here $(\lambda,\tilde{x})$ is moving relative to the stationary system (t,x). The difference only appears in the construction of the distribution functions in both g and f_0, which are presented in the following.

In the local moving frame of reference at interface $\xi = \text{constant}$, the Maxwellian distribution should have the form:

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{(K+2)/2}\exp\left\{-\lambda\left[\left(\tilde{u}-\tilde{U}\right)^2 + \left(\tilde{v}-\tilde{V}\right)^2 + \zeta^2\right]\right\},$$

where the averaged macroscopic fluid velocity $(\tilde{U},\tilde{V})$ is related to the fluid velocity (U,V) in the inertia frame of reference:

$$\begin{cases} \tilde{U} = (U-U_g)M/S - (V-V_g)L/S, \\ \tilde{V} = (U-U_g)L/S + (V-V_g)M/S. \end{cases} \tag{11}$$

Numerically, Eq. (9) is basically the same equation as the one we have solved in the Eulerian space, where $\tilde{u}, \tilde{v}$ are the particle velocity, and $\tilde{U}, \tilde{V}$ are the macroscopic velocity in the $\vec{n}$ and $\vec{t}$ directions. Then, the standard BGK–NS method [24] can be used to solve Eq. (9) to evaluate the time-dependent gas distribution function $f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ at the cell interface $\xi = \xi_{i+1/2}$. The detailed formulation of the gas-kinetic BGK–NS scheme for the Navier–Stokes solutions is given in [24]. Therefore, standing on the moving cell interface the fluxes can be explicitly obtained:

$$\begin{pmatrix} \mathscr{F}_\rho \\ \mathscr{F}_{\rho\tilde{u}} \\ \mathscr{F}_{\rho\tilde{v}} \\ \mathscr{F}_{\tilde{E}} \end{pmatrix}_{i+1/2,j} = \int \tilde{u} \begin{pmatrix} 1 \\ \tilde{u} \\ \tilde{v} \\ \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2 + \zeta^2) \end{pmatrix} f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta) \mathrm{d}\tilde{u}\,\mathrm{d}\tilde{v}\,\mathrm{d}\zeta. \tag{12}$$

Since different numerical cells can move with different grid velocity, in order to update the flow variables inside each time-dependent computational cell we need to update the conservative variables inside each cell in the common inertia frame of reference, i.e., the so-called Eulerian space. Therefore, we need to transfer the fluxes in Eq. (12) standing on the moving cell interface into the fluxes for the mass, momentum and energy transport of the inertia frame of reference. In other words, the above obtained gas distribution function $f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ and its mass flux across the moving cell interface $\tilde{u}f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta)$ will carry the mass, momentum and energy densities $(1, u, v, \frac{1}{2}(u^2 + v^2 + \zeta^2))$ in the inertia frame of reference. So, the time-dependent numerical flux in the Eulerian space in the $\vec{n}$ direction across the cell interface $\xi = C$ should be calculated as

$$\begin{pmatrix} \mathscr{F}_\rho \\ \mathscr{F}_m \\ \mathscr{F}_n \\ \mathscr{F}_E \end{pmatrix}_{i+1/2,j} = \int S\tilde{u} \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2 + v^2 + \zeta^2) \end{pmatrix} f(\xi_{i+1/2}, \eta_j, t, \tilde{u}, \tilde{v}, \zeta) \mathrm{d}\tilde{u}\,\mathrm{d}\tilde{v}\,\mathrm{d}\zeta. \tag{13}$$

In the above equation, the distribution function f is a scalar function, which is invariant under coordinate transformation, but the particle velocities $(\tilde{u}, \tilde{v})$ and (u, v) are defined differently in the different frame of reference. In order to evaluate the above flux integration, the easiest way is to write the (u, v) velocities in terms of $(\tilde{u}, \tilde{v})$. Based on the transformation (7), we have:

$$u = U_g + \frac{M\tilde{u} + L\tilde{v}}{S}, \quad v = V_g + \frac{-L\tilde{u} + M\tilde{v}}{S}.$$

Therefore, Eq. (13) becomes

$$\begin{pmatrix} \mathscr{F}_\rho \\ \mathscr{F}_m \\ \mathscr{F}_n \\ \mathscr{F}_E \end{pmatrix}_{i+1/2,j} = \begin{pmatrix} S\mathscr{F}_\rho \\ M\mathscr{F}_{\rho\tilde{u}} + L\mathscr{F}_{\rho\tilde{v}} + SU_g\mathscr{F}_\rho \\ -L\mathscr{F}_{\rho\tilde{u}} + M\mathscr{F}_{\rho\tilde{v}} + SV_g\mathscr{F}_\rho \\ (MU_g - LV_g)\mathscr{F}_{\rho\tilde{u}} + (LU_g + MV_g)\mathscr{F}_{\rho\tilde{v}} + S\mathscr{F}_{\tilde{E}} + \frac{S}{2}(U_g^2 + V_g^2)\mathscr{F}_\rho \end{pmatrix} \tag{14}$$

where $(\mathscr{F}_\rho, \mathscr{F}_{\rho\tilde{u}}, \mathscr{F}_{\rho\tilde{v}}, \mathscr{F}_{\tilde{E}})$ are given in Eq. (12). So, the fluxes relative to the moving cell interface in the Eulerian space is just a linear combination of the fluxes in the moving frame of reference due to the linear transformation between the inertia and moving space with relative velocity (U_g, V_g). Similarly, the fluxes at the cell interface $\eta =$ constant, i.e., $\mathbf{G}$, can be constructed as well.

With the above fluxes, the flow variables can be updated in each moving computational cell by

$$Q_{i,j}^{n+1} = Q_{i,j}^n + \frac{1}{\Delta\xi}\int_{t^n}^{t^{n+1}} \left(\mathbf{F}_{i-1/2,j} - \mathbf{F}_{i+1/2,j}\right)\mathrm{d}t + \frac{1}{\Delta\eta}\int_{t^n}^{t^{n+1}} \left(\mathbf{G}_{i,j-1/2} - \mathbf{G}_{i,j+1/2}\right)\mathrm{d}t, \tag{15}$$

where $Q = (\rho\Delta, \rho\Delta U, \rho\Delta V, \rho\Delta E)^{\mathrm{T}}$, $\mathbf{F} = (\mathscr{F}_\rho, \mathscr{F}_m, \mathscr{F}_n, \mathscr{F}_E)^{\mathrm{T}}$ given in Eq. (14), and $\mathbf{G}$ fluxes in the η-direction.

2.3. Adaptive grid method

If the moving grid velocity (U_g, V_g) is chosen according to the fluid velocity, such as the Lagrangian method with $(U_g, V_g) = (U, V)$, it is very hard to keep the mesh regularity. In almost all cases, the mesh will get tangled to generate a negative cell area Δ. In order to prevent this from happening, many mesh remedy techniques have been developed, such as ALE method. In this paper, we are going to use the adaptive mesh method [21] to overcome mesh-tangling problem. The function of this is two folds. Firstly, a smooth mesh will be generated before grid crashing. Secondly, the mesh can be automatically redistributed and concentrated on the region with high velocity or pressure gradients to improve the computational accuracy. However, when more grid points are concentrated in the shock region, the grid size becomes very small, as well as the time step. Therefore, the computational cost will be much increased. Also, since we do not resolve the shock structure, the order of accuracy cannot be improved by the mesh adaptation [26]. But, the absolute value of accuracy can be improved due to the small cell size. Certainly, it depends on the accuracy control in the interpolation between different meshes. In an early paper [13], an adaptive grid method coupling with gas-kinetic BGK scheme has been proposed. In the following, the mesh generation and adaptation are outlined.

In 2D, the widely used mesh generation techniques are based on the variational approaches. Let $\boldsymbol{x} = (x, y)$ and $\boldsymbol{\xi} = (\xi, \eta)$ denote the physical and computational coordinates. A coordinate mapping from the computational domain Ω_c to the physical domain Ω_p is given by

$$x = x(\xi, \eta), \quad y = y(\xi, \eta), \tag{16}$$

and the inverse map is

$$\xi = \xi(x, y), \quad \eta = \eta(x, y). \tag{17}$$

The specific map is obtained by minimizing a functional with the following form:

$$\tilde{E}[\xi, \eta] = \frac{1}{2} \int_{\Omega_c} \left(\tilde{\nabla}^{\mathrm{T}} x G_1 \tilde{\nabla} x + \tilde{\nabla}^{\mathrm{T}} y G_2 \tilde{\nabla} y\right) \mathrm{d}\xi \mathrm{d}\eta, \tag{18}$$

where $\tilde{\nabla} = (\partial/\partial\xi, \partial/\partial\eta)$ and G_1 and G_2 are symmetric positive definite matrices which are formally called monitor functions. The Euler–Lagrange equations can be obtained from the above equation:

$$\tilde{\nabla} \cdot \left(G_1 \tilde{\nabla} x\right) = 0, \quad \tilde{\nabla} \cdot \left(G_2 \tilde{\nabla} y\right) = 0. \tag{19}$$

Therefore, the mesh distribution in the physical space can be directly generated by solving (19). Numerically, in this paper a directional splitting monitor function [22] will be used, namely $G_1 = G_2 = \mathrm{diag}\{w_1, w_2\}$, where w_1, w_2 are defined by

$$w_1 = \left(1 + \alpha_1 |\psi|^2 + \beta_1 \left|\frac{\partial \psi}{\partial \xi}\right|^2\right)^{\frac{\gamma_1}{2}}, \quad w_2 = \left(1 + \alpha_2 |\psi|^2 + \beta_2 \left|\frac{\partial \psi}{\partial \eta}\right|^2\right)^{\frac{\gamma_2}{2}}. \tag{20}$$

In the above formula, ψ is the flow variable, such as velocity or pressure, α_1, α_2, β_1, β_2, γ_1, and γ_2 are some non-negative constants, and their optimum values depend on flow problems. A second-order central difference scheme can be used to discretize the mesh generation Eq. (19), and the difference equations can be solved by iterative methods, such as Gauss–Seidel iteration.

As the monitor functions (20) are assumed to be constants, i.e., $w_1 = w_2 = 1$, the Euler–Lagrange Eq. (19) become the well-known Laplace's equation, from which a smooth mesh can be constructed. The frequency to apply the adaptive mesh method depends on the physical problems solved. For example, in the first test case in Section 3, after a few testing calculations, we know precisely when the mesh will get severely distorted. Before that, a smoothing mesh method can be applied. After that, the mesh is smoothed every 200 time steps. However, for the examples (5) and (6), due to the requirement of accurate capturing of the moving shock and boundary layer, the adaptive grid technique is applied at every time step to redistribute the mesh concentration.

The purpose of this adaptive mesh generation is to regularize the mesh in the physical space which may be distorted due to the mesh movement. The newly generated mesh has only a small modification from the original mesh with a limited number of iterations in solving (19). After constructing the new mesh according to a

monitor function, we need to interpolate flow variables from the old to the newly obtained mesh. The conservative interpolation scheme proposed by Tang and Tang [21] for a finite volume formulation is used. More detailed formulation can be found in [13].

2.4. Numerical procedure

The numerical procedure for the gas-kinetic scheme on a moving and adaptive mesh can be summarize in the following. In some numerical examples, the adaptive mesh method is not used, and the corresponding step can be ignored. Also, all fluid quantities $(\rho, \rho U, \rho V, \rho E)$ and the geometric variables (A, B, M, L) are defined at each cell center. Any cell interface values are obtained through the average of the values in the adjacent cells.

Step 1: Initialize the conservative variables $(\rho, \rho u, \rho v, \rho E)$ and (A, B, L, M) at $t = \lambda = 0$ in the $x - y$ plane. Usually, ξ and η are the initial arc-length of their corresponding x- and y-coordinate lines. For example, for the rectangular domain, we take $\Delta\xi = \Delta x$, and $\Delta\eta = \Delta y$ when Δx and Δy are constants on the physical domain initially. Or, we can simply choose two constants to define $\Delta\xi$ and $\Delta\eta$. Then, (A, B, L, M) are determined according to the definition:

$$\begin{cases} A = \frac{\partial x}{\partial \xi}, \\ B = \frac{\partial y}{\partial \xi}, \\ L = \frac{\partial x}{\partial \eta}, \\ M = \frac{\partial y}{\partial \eta}. \end{cases} \tag{21}$$

Then, we calculate $(\Delta\rho, \Delta\rho u, \Delta\rho v, \Delta\rho E)$ inside each cell with $\Delta = AM - BL$.

Step 2: Construct or modify the grid using the adaptive grid method in Section 2.3, and update the conservative variables $(\rho, \rho u, \rho v, \rho E)$ inside each cell in the physical domain. Sequently, calculate (A, B, L, M) by Eq. (21) again and update $(\Delta\rho, \Delta\rho u, \Delta\rho v, \Delta\rho E)$ with the new $\Delta = AM - BL$. The re-distribution of conservative variables in this step is fully conservative.

Step 3: Given a grid velocity (U_g, V_g) at the center of each cell, such as the fluid velocity $\mathbf{q}$ in the gas-kinetic Lagrangian method, at the cell interface $\xi = \xi_{i+1/2}$ the grid velocity is calculated as

$$(U_g)_{i+1/2,j} = \frac{(U_g)_{i,j} + (U_g)_{i+1,j}}{2},$$
$$(V_g)_{i+1/2,j} = \frac{(V_g)_{i,j} + (V_g)_{i+1,j}}{2},$$

which are basically the average of the velocities from the neighboring cells. Then, based on (14) the numerical fluxes are calculated across the moving interface. At the same time, the center of the cell, such as $(x^n_{i,j}, y^n_{i,j})$, moves to a new location through

$$\begin{cases} x^{n+1}_{i,j} = x^n_{i,j} + U_g(t^{n+1} - t^n), \\ y^{n+1}_{i,j} = y^n_{i,j} + V_g(t^{n+1} - t^n). \end{cases} \tag{22}$$

The location of cell vertex is updated by averaging the centers of four neighboring cells.

Step 4: With the new mesh location x^{n+1}, y^{n+1}, calculate (A, B, L, M) and Δ using (21). Then, update the conservative variables by the finite volume scheme (15) in the newly moved cells. It can be shown that the use of (21) to calculate (A, B, L, M) is the same as the geometrical conservation law (3), such as the update of A:

$$A^{n+1}_{i,j} = \left(\frac{\partial x}{\partial \xi}\right)^{n+1}_{i,j} = \frac{x^{n+1}_{i+1/2,j} - x^{n+1}_{i-1/2,j}}{\Delta\xi} = \frac{x^n_{i+1/2,j} + (U_g)^n_{i+1/2,j}\Delta\lambda^n - x^n_{i-1/2,j} - (U_g)^n_{i-1/2,j}\Delta\lambda^n}{\Delta\xi}$$
$$= \frac{x^n_{i+1/2,j} - x^n_{i-1/2,j}}{\Delta\xi} + \frac{\Delta\lambda^n}{\Delta\xi}\left((U_g)^n_{i+1/2,j} - (U_g)^n_{i-1/2,j}\right) = A^n_{i,j} + \frac{\Delta\lambda^n}{\Delta\xi}\left((U_g)^n_{i+1/2,j} - (U_g)^n_{i-1/2,j}\right).$$

Similarly, it is true for B, L, and M.

Then, go to step (2) to repeat the above process until the output time.

3. Numerical experiments

As mentioned earlier, the main advantage of the current method is that the fluid system under unified coordinate is coupled with the geometrical conservation laws. Mathematically, the unified system provides a complete enlarged system, which avoids the difficulties in other moving mesh methods to implement the geometrical conservation separately based on the physical intuition. Also, different from the Eulerian and Lagrangian methods, the choice of grid velocity (U_g, V_g) becomes a new degree of freedom in the current model. The proper choice of the grid speed depends on the specific applications to get the optimum accuracy in the numerical solution. For example, the grid velocity can be the fluid velocity in the free surface flow, or follow the oscillating angular velocity for a pitching airfoil. In the approach by Hui et al. [10], to preserve the grid angle has been used to get the local value h in the determination of grid velocity (hU, hV). In this section, the BGK–NS scheme on a moving mesh will be tested on many examples. Different choice of the grid velocity will be used. The numerical results are compared with the exact solutions, experimental data, and the available solutions obtained by others.

Case (1) *Free surface flow*

The current approach with the choice of $(U_g, V_g) = (U, V)$ becomes a purely gas-kinetic Lagrangian method even though it is solving the viscous governing equations. The use of the fluid velocity as the grid speed can naturally capture the free surface. The case we are going to study is the dam break problem, where a column of water is released by removing a vertical diaphragm. This becomes a standard benchmark problem due to its simple geometry and the available experimental measurement [12]. The initial configuration is shown in the upper left picture in Fig. 2. In this example, a rectangular column of water in hydrostatic equilibrium is confined initially between two vertical walls. The water column is 7.0 units high and 3.5 units wide. The gravity is acting downward with 0.05 unit magnitude.

After the diaphragm eruption at time $t = 0$, the water is pushing out and moves freely along a dry horizontal floor. The measured quantities include the water wave front location L on the floor. In our calculation, 40×20 rectangular mesh points are initially employed in a domain $0 \leqslant x \leqslant 3.5$, $0 \leqslant y \leqslant 7.5$. Fig. 2 presents the moving mesh at 4 subsequent times. Since the grid speed is equal to the fluid velocity, the mesh distribution is the same as the water distribution. In a purely Lagrangian simulation, the mesh is easily tangled at a later time. Since we have used a mesh smoothing technique through the mesh adaptive method with a constant monitor function to equally distribute the mesh, the moving mesh becomes generally smooth all over the domain even though the mesh adaptation steps are applied only a few times.

Fig. 3 shows the water tip location versus time for both simulation and experimental measurement. The non-dimensional time in the horizontal coordinate is normalized by $t\sqrt{2g/W}$.

Case (2) *2D model of a pitching airfoil*

This is about the flow passing through a pitching airfoil. The airfoil undergoes pitching oscillations around a point on the chord with one quarter length from the leading edge. In the current case, a NACA 0012 airfoil was used and the chord was placed at the x axis initially. The free stream velocity U_∞ is parallel to the x axis with a Mach number $M = 0.755$. The oscillation cycle is defined by

$$\alpha = \alpha_m + \alpha_0 \sin \omega t, \quad \text{with} \quad \alpha_m = 0.016^\circ \quad \text{and} \quad \alpha_0 = 2.51^\circ,$$

where ω is the circular pitch frequency, c is chord length, and $\omega c/2U_\infty = 0.0814$. In many computations, the mesh is fixed and the free stream U_∞ velocity is being rotated in the opposite direction as the airfoil pitching. However, in the present calculation, the whole mesh around the airfoil is moving with the airfoil and the grid velocity (U_g, V_g) is calculated according to the angular velocity of the pitching airfoil and the distance between the grid point and oscillating center. The oscillating center is at the origin. Since only mesh is rotating, all flow variables at ∞, such as U_∞, are unchanged in the current Eulerian space. In our calculation, a U-type mesh with 178×44 grid points on a span of eight chord distance is used. Fig. 4 shows two instant meshes near the airfoil at different angles of attack.

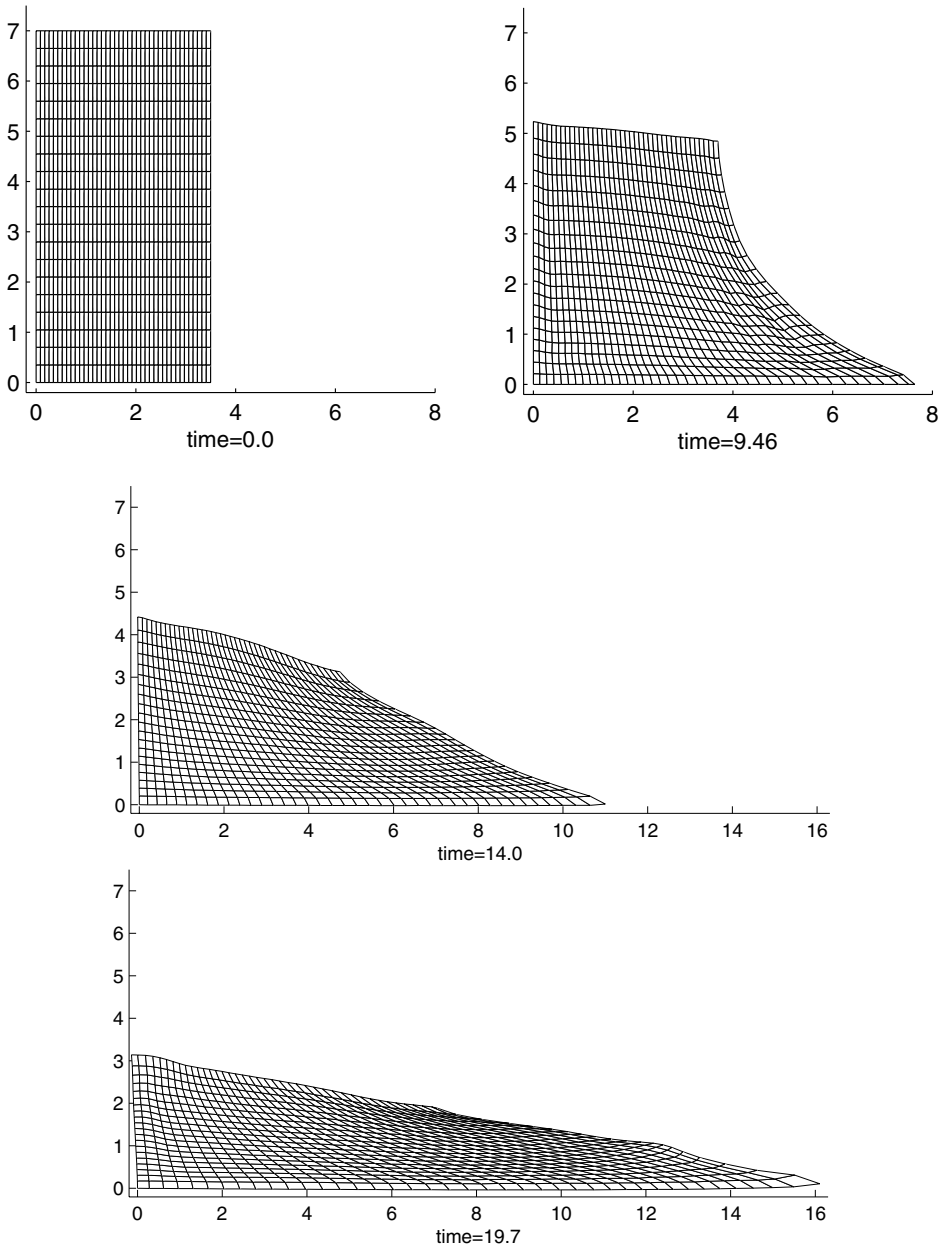


Fig. 2. Moving mesh at time $t = 0, 9.46, 14.0, 19.7$.

Fig. 5 presents the lift coefficient C_l versus instantaneous angle of attack α. The numerical solutions give good agreements with the experimental data. Fig. 6 presents the detailed pressure distribution C_p around the surface of the airfoil at different angle of attack α. Also, good agreement between the numerical and experimental results are obtained. Since the grid velocity in the current unified gas-kinetic scheme can be chosen freely, it may have advantage in applying this method in the biology insect flight computations where the mesh can follow the complicated wing movement.

Case (3) *Shock reflection inside a channel with a ramp*

In order to preserve the grid angles [10], the grid speed $(U_g, V_g) = (hU, hV)$ can be obtained by solving the following equation for h:

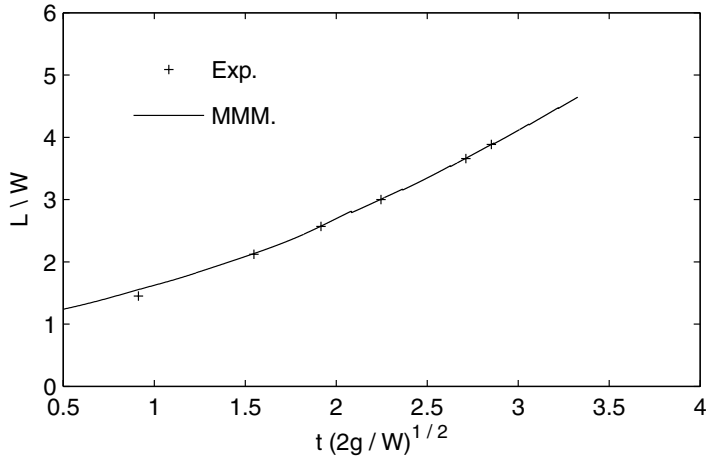


Fig. 3. The time dependent location of water front. Experimental measurement versus the calculation from moving mesh method (MMM).

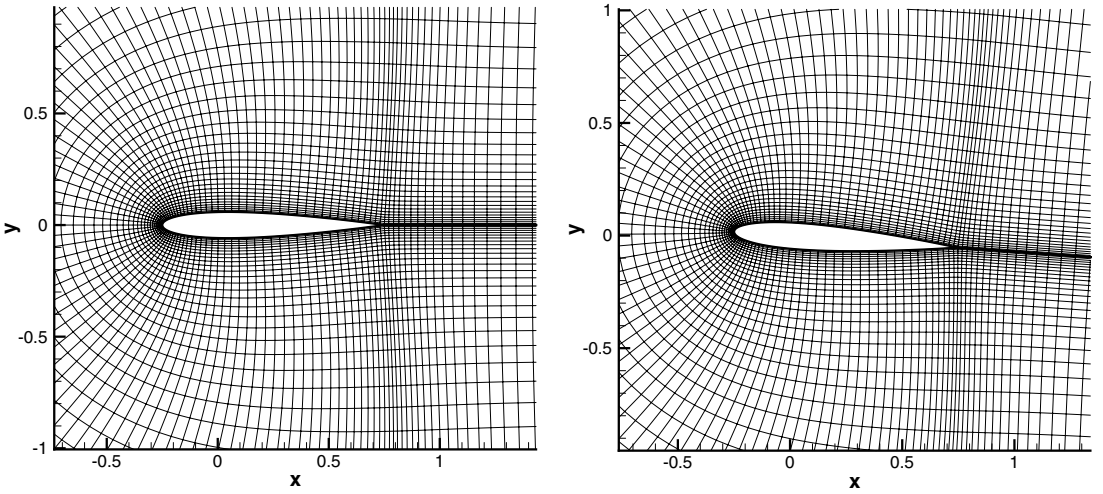


Fig. 4. Moving meshes near NACA 0012 airfoil at two different angles of attack: $\alpha = \alpha_m$ (left) and $\alpha = \alpha_m + \alpha_0$ (right).

$$\frac{\partial}{\partial\lambda}\left[\frac{\nabla\xi}{|\nabla\xi|}\cdot\frac{\nabla\eta}{|\nabla\eta|}\right] = 0,$$

where $\nabla\xi = (M, -L)/\Delta$, $\nabla\eta = (-B, -A)/\Delta$, and θ is the flow angle: $U = q\cos\theta$, $V = q\sin\theta$, $q = \sqrt{U^2 + V^2}$. Let $\tilde{g} = \ln(hq)$, then the above equation becomes:

$$S^2(A\sin\theta - B\cos\theta)\frac{\partial\tilde{g}}{\partial\xi} + T^2(M\cos\theta - L\sin\theta)\frac{\partial\tilde{g}}{\partial\eta} = S^2\left(B\frac{\partial\cos\theta}{\partial\xi} - A\frac{\partial\sin\theta}{\partial\xi}\right) - T^2\left(M\frac{\partial\cos\theta}{\partial\eta} - L\frac{\partial\sin\theta}{\partial\eta}\right). \tag{23}$$

Numerically, Eq. (23) can be solved by an iterative method to get h at different grid point. The detail description was given in [10].

In the current supersonic flow of $M = 1.8$ passing through a ramp in a channel, Eq. (23) is used to calculate h, then determine the grid velocity. Here the ramp with 15° is placed at the lower wall between $x = 0.5$ and $x = 1.0$. A computational grid with 180×50 grid points in the physical domain $\{0 \leqslant x \leqslant 3.6,\ 0 \leqslant y \leqslant 1.0\}$

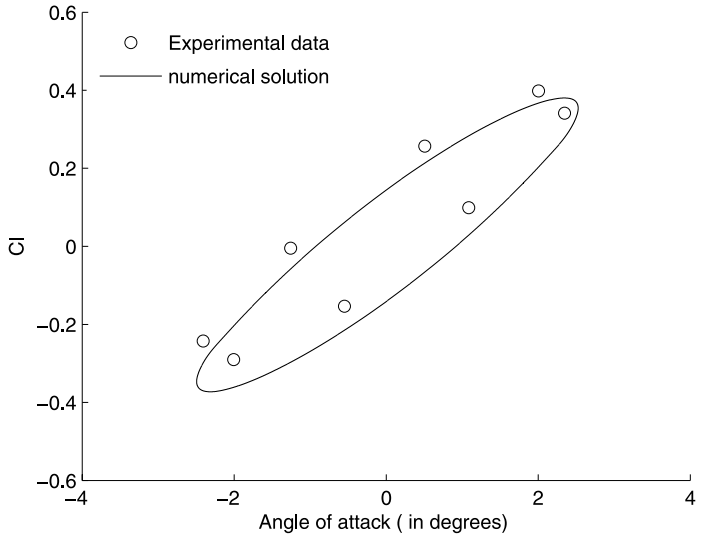


Fig. 5. Lift coefficients versus angle of attack in the airfoil pitching problem.

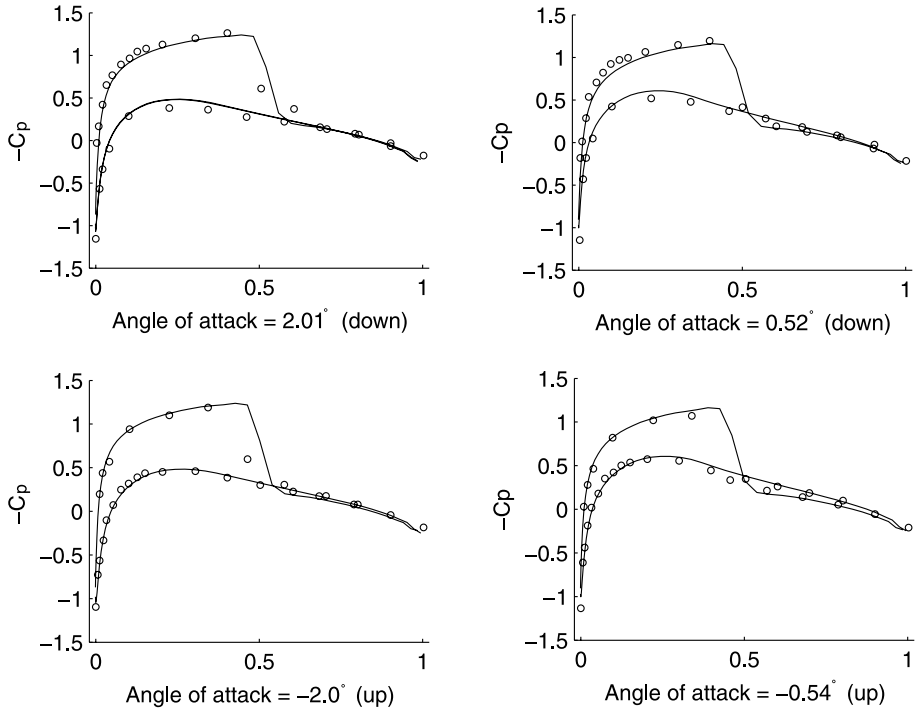


Fig. 6. Pressure distributions at different angle of attack.

is initially generated. An initial flow data with $(p, \rho, M, \theta) = (1, 1, 1.8, 0)$ is imposed inside each cell, as well as at the left boundary. Reflection boundary conditions are used at the top and bottom walls. When h is chosen according to Eq. (23), the mesh will automatically preserve the grid angle, which is shown in Fig. 7. Fig. 8 presents pressure and Mach number distributions after the steady state solution is obtained.

Case (4) *Viscous solution above an oscillating wall*

This is called Stokes' second problem, which considers fluid motion above an infinite flat plate which executes sinusoidal oscillations parallel to itself. This problem has been simulated earlier by the purely adaptive

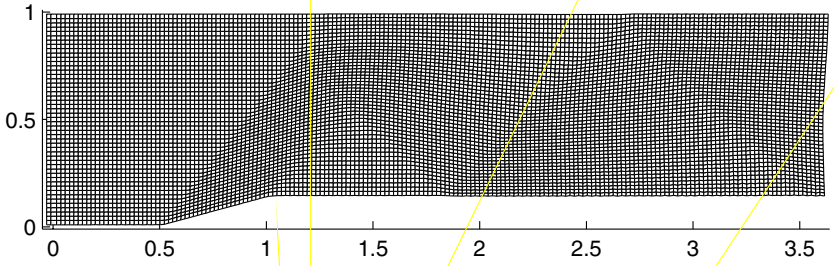
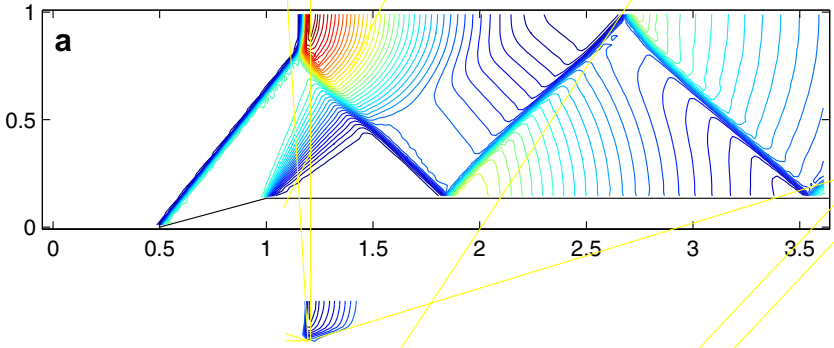


Fig. 7. Computational mesh due to the implementation of grid angle preservation.



mesh method [13]. The fluid above the plate is initially stationary. The governing equation of velocity U in the x-direction is

$$\frac{\partial U}{\partial t} = \nu \frac{\partial^2 U}{\partial y^2}, \tag{24}$$

with the boundary conditions

$$U_{\text{wall}}(0,t) = U_0 \cos \omega t, \quad U(\infty,t) = 0.$$

The exact solution for the above problem is,

$$U(y,t) = U_0 \mathrm{e}^{-y\sqrt{\omega/2\nu}} \cos\left(\omega t - y\sqrt{\frac{\omega}{2\nu}}\right). \tag{25}$$

At $y = 4\sqrt{\nu/\omega}$, the amplitude of U is equal to $U_0 \exp(-4/\sqrt{2}) = 0.05U_0$, which means that the influence from the wall is confined within a distance of order $\delta \sim 4\sqrt{\nu/\omega}$. Since the gas-kinetic scheme solves the compressible Navier–Stokes equations, in order to simulate the above incompressible limiting solution the Mach number for the compressible flow takes a small value, i.e., $M = 0.15$. The kinematic viscosity coefficient takes a value $\nu = 0.00046395$, and a mesh size with 10×70 grid points is used.

In the current calculation, we have used two ways to determine the mesh velocity. In the first case, we used the purely Lagrangian method for the viscous solution, where the grid velocity follows the fluid one. Due to

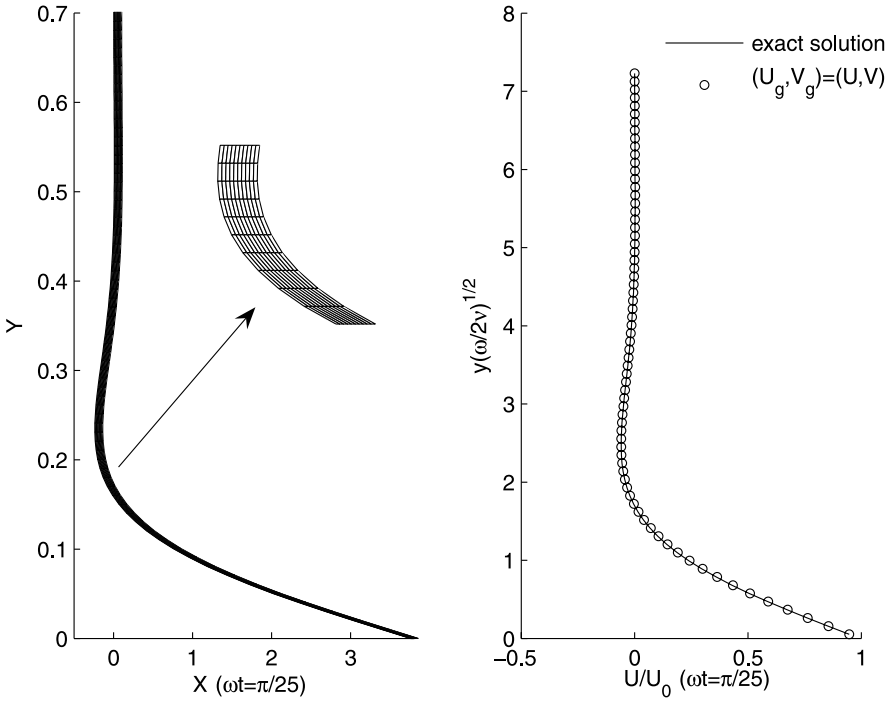


Fig. 9. Lagrangian gas-kinetic scheme for viscous flow. Mesh (left) and velocity (right) distributions at time $\omega t = \pi/25$.

large velocity shear in the boundary layer, the Lagrangian method will stretch the mesh severely. This is the main reason why nobody really used Lagrangian method for the viscous flow computation. However, it is still theoretically interesting to see the solution using gas-kinetic Lagrangian method. Fig. 9 shows the mesh (left)

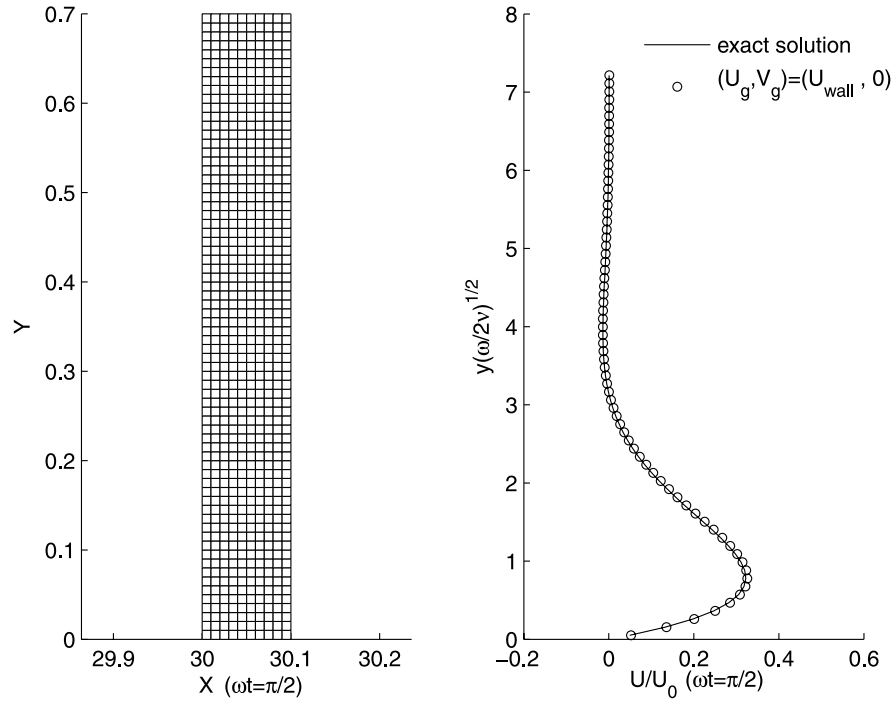


Fig. 10. Gas-kinetic scheme for viscous flow with $(U_g, V_g) = (U_{\text{wall}}, 0)$. Mesh (left) and velocity (right) distributions at time $\omega t = \pi/2$.

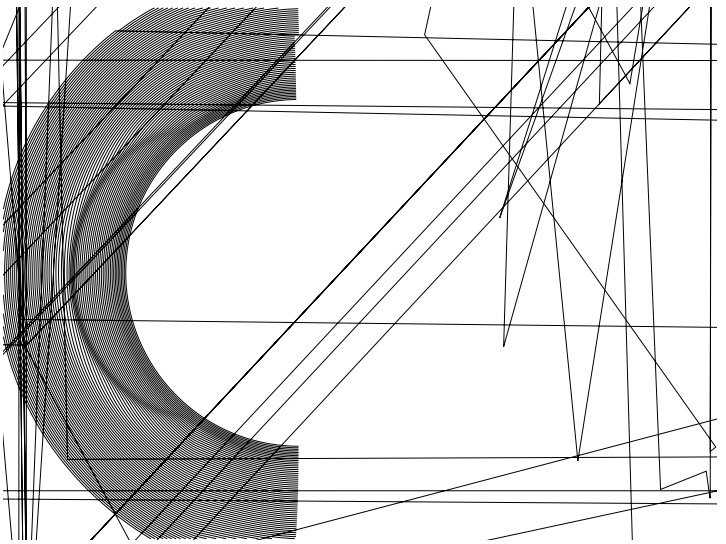
and velocity (right) at time $\omega t = \pi/25$ when the mesh follows the fluid velocity. Even though the mesh has been stretched greatly, it is surprising that the numerical solution is very close to the exact solution. This proves the robustness and accuracy of the current kinetic scheme for the viscous computation. With the time increasing, the mesh will be stretched further until 70 grid points are not enough to follow the time increasing velocity arc-length. Eventually, the computation will stop. If the mesh velocity at all grid points follow the wall velocity, such as $(U_g, V_g) = (U_{\text{wall}}, 0)$, the mesh will not get tangled. Fig. 10 shows the mesh and simulation results at $\omega t = \pi/2$.

Case (5) *Shock wave interaction with a rotating cylinder*

An incident shock wave with Mach number $M_s = 1.19$ from the left impinges on an anti-clockwise rotating cylinder. The test gas is air at pressure $P_{-\infty} = 100$ kPa and temperature $T_{-\infty} = 298$ K, respectively. In the current computation, non-slip boundary condition is imposed around the surface of the rotating cylinder. The dynamical viscosity coefficient μ takes the power law:

$$\frac{\mu}{\mu_{-\infty}} = \left(\frac{T}{T_{-\infty}}\right)^n = \left(\frac{1/\lambda}{1/\lambda_{-\infty}}\right)^n, \quad n = 0.666,$$

where $\lambda = \rho/2p$, $\lambda_{-\infty} = \rho_{-\infty}/2p_{-\infty}$, and $\mu_{-\infty} = 1.716 \times 10^{-5}$ Ns/m^2. The cylinder has a radius $R = 0.07$ m and the surface rotating velocity has a value to be equivalent to Mach number $M = 0.3$. Besides the mesh velocity follows the rotating cylinder with a value $r\omega$, where r is the distance from the rotating center to the mesh point, an adaptive mesh method is also used to concentrate the mesh in the high velocity gradient region [13]. The solutions around the shock front and inside the boundary layer are captured accurately due to the mesh concentration. Numerical mesh at times $t = 0.106$, 0.179 are shown in Fig. 11. Fig. 12 shows the sequential pressure contours after shock reflection from the cylindrical surface. Because of the viscous effect, the locations of the triple points on the upper and lower surfaces are different, as well as the length of Mach stem. The Mach stem is longer on the lower surface where the shock wave propagates with the same direction as the rotating surface. On the contrary, on the upper surface the Mach stem is suppressed due to their opposite moving directions. Although accurate measurement of shock wave over a moving body is rather difficult, the different shock wave distribution on both upper and lower surfaces has been observed experimentally. In [18], the experimental observation showed that the length of the Mach stem on the lower surface is indeed longer than that on the upper surface, which is consistent with the current computational result. For the inviscid flow computation, the solution is irrelevant with the rotation of the cylinder due to the implementation of slip boundary condition.



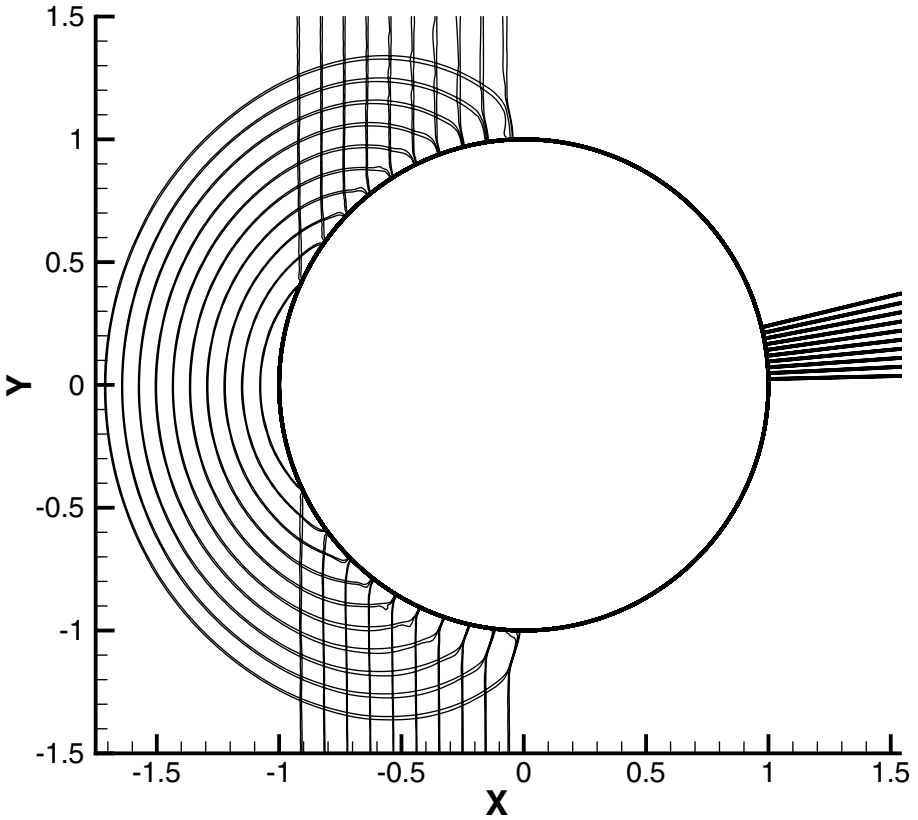


Fig. 12. Sequential pressure contours after shock reflection from the rotating cylinder.

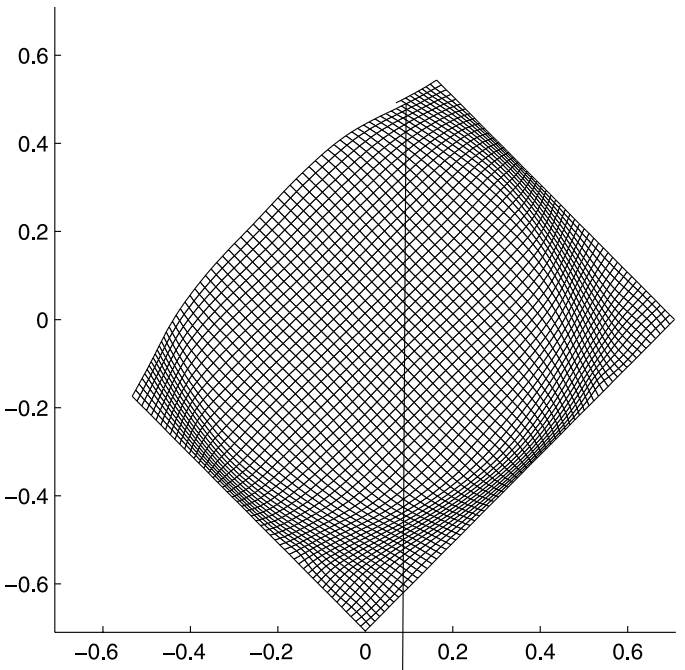


Fig. 13. Moving and adaptive grids at time t $= \frac{3}{4}$p for the rotating cavity case.

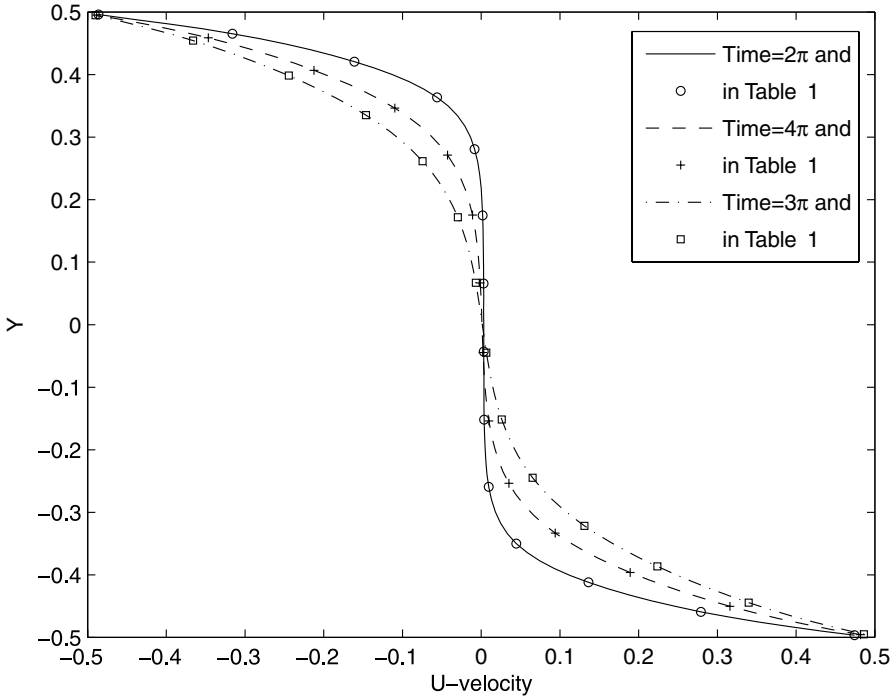


Fig. 14. Velocity U profiles along the central vertical line at $Re = 1000$ after one, two and three full rotations.

Table 1
U-velocity distributions along vertical central line at $Re = 1000$

Time $= 2\pi$		Time $= 4\pi$		Time $= 6\pi$	
y	U	y	U	y	U
−0.49674	0.47419	−0.49580	0.48181	−0.49513	0.48611
−0.45919	0.27910	−0.45029	0.31605	−0.44439	0.33977
−0.41211	0.13619	−0.39639	0.18935	−0.38667	0.22391
−0.35012	0.04468	−0.33326	0.09404	−0.32162	0.13113
−0.25932	0.00967	−0.25374	0.03519	−0.24469	0.06532
−0.15179	0.00382	−0.15394	0.00975	−0.15151	0.02615
−0.04298	0.00329	−0.04441	0.00235	−0.04476	0.00663
0.06586	0.00299	0.06659	−0.00114	0.06698	−0.00688
0.17473	0.00177	0.17518	−0.01107	0.17172	−0.02974
0.28056	−0.00831	0.27106	−0.04276	0.26130	−0.07437
0.36354	−0.05603	0.34621	−0.10971	0.33513	−0.14631
0.42069	−0.16095	0.40677	−0.21226	0.39825	−0.24436
0.46533	−0.31610	0.45867	−0.34656	0.45427	−0.36586
0.49594	−0.48614	0.49525	−0.48838	0.49474	−0.48976

Case (6) *Rotating cavity flow*

The cavity flow inside a square cylinder with upper moving boundary has been calculated by many authors and becomes one of the benchmark test case for the incompressible Navier–Stokes equations [8]. Here, we design a new test case, where the whole 2D square cavity with boundary length 1 starts to rotate around its center at time $t = 0$ with a constant angular velocity $\omega = 1$, where the viscosity coefficient is set to a value with the Reynolds number $Re = 1000$ for velocity difference $\Delta U = 1$ and $L = 1$. Since we are using a compressible code to calculate the incompressible solution, the initial Mach number according to the rotating velocity

at the cavity surface is $M = 0.15$. A mesh with 65×65 grid points is used in the current computation. The computational mesh is rotating with the cavity, at the same time adapted according to the velocity gradients. Fig. 13 shows the mesh in the physical domain at time$= \frac{3}{4}\pi$. Obviously, grid points are concentrated in regions close to the boundaries with large velocity gradients. Fig. 14 displays the velocity U profiles along the central vertical line after one, two, and three rotations $\omega t = 2\pi$, 4π, and 6π. The specific velocity values are listed in Table 1 for future reference. Since this is an unsteady problem, the U-velocity along vertical central line depends on time in the early transition period.

4. Conclusion

In this paper, the gas-kinetic BGK equation is transformed into a moving frame of reference and a unified numerical scheme for the viscous solution is developed. This is a finite volume gas-kinetic scheme on a moving grid in the Eulerian space and the mesh velocity can be properly chosen to capture flow movement with moving boundaries. The Eulerian and Lagrangian methods are two limiting cases for the current scheme. In order to increase the robustness and accuracy of the current method, the mesh adaptation is also implemented in the current method, which not only remedies the distorted mesh, but also concentrates the mesh in the high gradient regions. The current unified gas-kinetic method has been applied to many flow problems, such as the free surface flow and Mach reflection inside a channel, where both inviscid and viscous solutions have been accurately obtained. Also, for the first time, the Lagrangian gas-kinetic scheme has been used in the viscous boundary layer calculation. The great advantage of the current scheme with a variable mesh velocity is that the gas-kinetic equation and geometrical conservation laws are combined as a unified system. The physical and geometrical variables can be updated simultaneously. Due to the relaxation term in the gas-kinetic model, the numerical treatment of the complicated viscous terms in the Navier–Stokes equations under the unified coordinate transformation (see appendix) can be avoided.

Acknowledgements

The authors would like to thank reviewers for their constructive comments, and to Dr. Z.W. Li and Q.B. Li for their help and discussion on the moving mesh method. The work described in this paper was substantially supported by grants from the Research Grants Council of the Hong Kong Special Administrative region, China (Project No. HKUST6210/05E and 6214/06E).

Appendix. The Navier–Stokes equations under unified coordinate transformation

The flow equations in 2D Eulerian space can be expressed as

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = \frac{\partial E_v}{\partial x} + \frac{\partial F_v}{\partial y}, \tag{26}$$

where

$$Q = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, \quad E = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{pmatrix}, \quad E_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix}, \quad F = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \end{pmatrix}, \quad F_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{yx} + v\tau_{yy} - q_y \end{pmatrix}.$$

For a Newtonian fluid,

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x} + \kappa\nabla\cdot\vec{V}, \quad \tau_{yy} = 2\mu\frac{\partial v}{\partial y} + \kappa\nabla\cdot\vec{V}, \quad \tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \tag{27}$$

and

$$q_x = -\frac{\mu c_p}{P_r}\frac{\partial T}{\partial x}, \quad q_y = -\frac{\mu c_p}{P_r}\frac{\partial T}{\partial y}. \tag{28}$$

With the following transformation between the physical (t, x, y) and computational space (λ, ξ, η),

$$\begin{cases} \mathrm{d}t = \mathrm{d}\lambda \\ \mathrm{d}x = x_\lambda \mathrm{d}\lambda + x_\xi \mathrm{d}\xi + x_\eta \mathrm{d}\eta \\ \mathrm{d}y = y_\lambda \mathrm{d}\lambda + y_\xi \mathrm{d}\xi + y_\eta \mathrm{d}\eta, \end{cases} \tag{29}$$

we can get the inverse transformation,

$$\begin{pmatrix} \mathrm{d}\lambda \\ \mathrm{d}\xi \\ \mathrm{d}\eta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} \mathrm{d}t \\ \mathrm{d}x \\ \mathrm{d}y \end{pmatrix}. \tag{30}$$

The N–S Eq. (26) in the computational space (λ, ξ, η) become:

$$\frac{\partial \overline{Q}}{\partial \lambda} + \frac{\partial \overline{E}}{\partial \xi} + \frac{\partial \overline{F}}{\partial \eta} = \frac{\partial \overline{E_v}}{\partial \xi} + \frac{\partial \overline{F_v}}{\partial \eta}, \tag{31}$$

where

$$\begin{aligned} \overline{Q} &= \Delta Q, \\ \overline{E} &= \Delta(\xi_t Q + \xi_x E + \xi_y F), \\ \overline{F} &= \Delta(\eta_t Q + \eta_x E + \eta_y F), \\ \overline{E_v} &= \Delta(\xi_x E_v + \xi_y F_v), \\ \overline{F_v} &= \Delta(\eta_x E_v + \eta_y F_v), \end{aligned} \tag{32}$$

and $\Delta = \partial(x, y)/\partial(\xi, \eta) = x_\xi y_\eta - x_\eta y_\xi$.

For a system with mesh moving velocity (U_g, V_g), we have:

$$\begin{pmatrix} 1 & 0 & 0 \\ x_\lambda & x_\xi & x_\eta \\ y_\lambda & y_\xi & y_\eta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ U_g & A & L \\ V_g & B & M \end{pmatrix}. \tag{33}$$

Therefore, we get

$$\begin{pmatrix} 1 & 0 & 0 \\ \xi_t & \xi_x & \xi_y \\ \eta_t & \eta_x & \eta_y \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} \Delta & 0 & 0 \\ -I_g & M & -L \\ -J_g & -B & A \end{pmatrix}, \tag{34}$$

where $\Delta = AM - BL$, $I_g = MU_g - LV_g$, and $J_g = AV_g - BU_g$. Hence, the variables in Eq. (31) are:

$$\begin{aligned} \overline{Q} &= \Delta Q, \\ \overline{E} &= (-I_g Q + ME - LF), \\ \overline{F} &= (-J_g Q - BE + AF), \\ \overline{E_v} &= (ME_v - LF_v), \\ \overline{F_v} &= (-BE_v + AF_v). \end{aligned} \tag{35}$$

The viscous shear stresses given by Eq. (27) with Stokes' hypothesis $\kappa = -\frac{2}{3}\mu$ become the following in the transformed computational space:

$$\begin{aligned}
\tau_{xx} &= \frac{\mu}{\Delta}\left[\frac{4}{3}(Mu_\xi - Bu_\eta) - \frac{2}{3}(-Lv_\xi + Av_\eta)\right],\\
\tau_{yy} &= \frac{\mu}{\Delta}\left[\frac{4}{3}(-Lv_\xi + Av_\eta) - \frac{2}{3}(Mu_\xi - Bu_\eta)\right],\\
\tau_{xy} &= \frac{\mu}{\Delta}(-Lu_\xi + Au_\eta + Mv_\xi - Bv_\eta),
\end{aligned} \tag{36}$$

and the heat conduction terms in Eq. (28) are:

$$\begin{aligned}
q_x &= -\frac{\mu c_p}{p_r \Delta}(MT_\xi - BT_\eta),\\
q_y &= -\frac{\mu c_p}{p_r \Delta}(-LT_\xi + AT_\eta).
\end{aligned} \tag{37}$$

In the computational space, the viscous flux vector $\overline{E_v}$ and $\overline{F_v}$ are

$$\overline{E}_v = \frac{\mu}{\Delta}\begin{pmatrix}
0\\
\left(\frac{4}{3}M^2 + L^2\right)u_\xi - \frac{1}{3}MLv_\xi + \left(-\frac{4}{3}MB - AL\right)u_\eta + \left(-\frac{2}{3}AM + BL\right)v_\eta\\
-\frac{1}{3}MLu_\xi + \left(M^2 + \frac{4}{3}L^2\right)v_\xi + \left(MA - \frac{2}{3}BL\right)u_\eta + \left(-MB - \frac{4}{3}AL\right)v_\eta\\
\frac{1}{2}\left(\frac{4}{3}M^2 + L^2\right)(u^2)_\xi + \frac{1}{2}\left(M^2 + \frac{4}{3}L^2\right)(v^2)_\xi - \frac{1}{3}ML(uv)_\xi +\\
\frac{c_p(M^2+L^2)}{P_r}T_\xi + \frac{1}{2}\left(-\frac{4}{3}MB - AL\right)(u^2)_\eta + \frac{1}{2}\left(-MB - \frac{4}{3}AL\right)(v^2)_\eta +\\
\left(MA - \frac{2}{3}BL\right)vu_\eta + \left(BL - \frac{2}{3}MA\right)uv_\eta + \frac{c_p(-MB-AL)}{P_r}T_\eta
\end{pmatrix}, \tag{38}$$

and

$$\overline{F}_v = \frac{\mu}{\Delta}\begin{pmatrix}
0\\
\left(-\frac{4}{3}MB - AL\right)u_\xi + \left(MA - \frac{2}{3}BL\right)v_\xi + \left(\frac{4}{3}B^2 + A^2\right)u_\eta - \frac{1}{3}BAv_\eta\\
\left(BL - \frac{2}{3}MA\right)u_\xi + \left(-MB - \frac{4}{3}AL\right)v_\xi - \frac{1}{3}ABu_\eta + \left(B^2 + \frac{4}{3}A^2\right)v_\eta\\
\frac{1}{2}\left(-\frac{4}{3}MB - AL\right)(u^2)_\xi + \frac{1}{2}\left(-MB - \frac{4}{3}AL\right)(v^2)_\xi + \left(AM - \frac{2}{3}BL\right)uv_\xi +\\
\left(BL - \frac{2}{3}AM\right)vu_\xi + \frac{c_p(-MB-AL)}{P_r}T_\xi + \frac{1}{2}\left(\frac{4}{3}B^2 + A^2\right)(u^2)_\eta +\\
\frac{1}{2}\left(B^2 + \frac{4}{3}A^2\right)(v^2)_\eta - \frac{1}{3}AB(uv)_\eta + + \frac{c_p(B^2+A^2)}{P_r}T_\eta
\end{pmatrix}. \tag{39}$$

References

[1] B.N. Azarenok, T. Tang, Second-order Gounov-type scheme for reactive flow calculations on moving meshes, J. Comput. Phys. 206 (2005) 48–80.
[2] J.T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA J. 28 (1990) 1381–1388.
[3] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases I: Small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511–525.
[4] W.M. Cao, W.Z. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.
[5] S. Chapman, T.G. Cowling, The Mathematical Theory of Non-uniform Gases, Cambridge University Press, 1990.
[6] X.Y. Chen, G.C. Zha, Fully coupled fluid-structural interactions using an efficient high resolution upwind scheme, J. Fluids Struct. 20 (2005) 1105–1125.
[7] K.A. Hoffmann, S.T. Chiang, Computational fluid dynamics, third ed., vol. 2, Engineering Education System, Wichita, Kansas, pp. 21–46.
[8] U. Ghia, K.N. Ghia, C.T. Shin, High-Resolutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[9] W. Huang, Mathematical principles of anisotropic mesh adaptation, Commun. Comput. Phys. 1 (2006) 276–310.
[10] W. H Hui, P.Y. Li, Z.W. Li, A unified coordinated system for solving the two-dimensional Euler equations, J. Comput. Phys. 153 (1999) 596–637.
[11] W.H. Hui, G.P. Zhao, Capturing contact discontinuities using the unified coordinates, Proc. Second MIT Conf. on Computational Fluid and Solid Mechanics (2003) 2000–2003.

[12] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free surface boundaries, J. Comput. Phys. 39 (1981) 210–225.
[13] C.Q. Jin, K. Xu, An adaptive grid method for two dimensional viscous flows, J. Comput. Phys. (2006), doi:10.1016/j.jcp.2006.01.041.
[14] K. Lipnikov, M. Shashkov, The error minimization based strategy for moving mesh methods, Commun. Comput. Phys. 1 (2006) 53–80.
[15] V. Ramesh, S.M. Deshpande, Least squares kinetic upwind method on moving grids for unsteady Euler computations, Comput. Fluids 30 (2001) 621–641.
[16] B.A. Robinson, J.T. Batina, H.T.Y. Yang, Aeroelastic analysis of wings using the Euler equations with a deforming mesh, J. Aircraft 28 (1991) 781–788.
[17] W. Shyy, H.S. Udaykumar, M.M. Rao, R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, Taylor and Francis, Washington, DC, 1996.
[18] M. Sun, K. Yada, G. Jagadeesh, O. Onodera, T. ogawa, K. Takayama, A study of shock wave interaction with a rotating cylinder, Shock Waves 12 (2003) 479–485.
[19] M. Sun, T. Saito, P.A. Jacobs, E.V. Timofeev, K. ohtani, K. Takayama, Axisymmetric shock wave interaction with a cone: A benchmark test, Shock Waves 14 (5–6) (2003) 313–331.
[20] J.M. Stockie, J.A. Mackenzie, R.D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, SIAM J. Sci. Comput. 22 (2001) 1791–1813.
[21] H.Z. Tang, T. Tang, Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[22] H.Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys. 1 (2006) 656–676.
[23] H.M. Tsai, A.S.F. Wong, J. Cai, F. Liu, Unsteady flow calculations with a parallel multiblock moving mesh algorithm, AIAA J. 39 (2001) 1021–1029.
[24] 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.
[25] K. Xu, M.L. Mao, L. Tang, A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow, J. Comput. Phys. 203 (2005) 405–421.
[26] N.K. Yamaleev, M.H. Carpenter, On accuracy of adaptive grid methods for captured shocks, J. Comput. Phys. 181 (2002) 280–316.