Journal of Computational Physics 228 (2009) 3154–3171

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Remapping-free ALE-type kinetic method for flow computations

Guoxi Ni [a,*], Song Jiang [a], Kun Xu [b]

[a] LCP, Institute of Applied Physics and Computational Mathematics, P.O. Box 8009, Beijing 100088, China
[b] Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 15 July 2008
Received in revised form 28 December 2008
Accepted 14 January 2009
Available online 26 January 2009

MSC:
65M06
76P05
76T05

Keywords:
Relative velocity
MMBGK scheme
ALE
Euler and Navier–Stokes equations

ABSTRACT

Based on the integral form of the fluid dynamic equations, a finite volume kinetic scheme with arbitrary control volume and mesh velocity is developed. Different from the earlier unified moving mesh gas-kinetic method [C.Q. Jin, K. Xu, An unified moving grid gas-kinetic method in Eulerian space for viscous flow computation, J. Comput. Phys. 222 (2007) 155–175], the coupling of the fluid equations and geometrical conservation laws has been removed in order to make the scheme applicable for any quadrilateral or unstructured mesh rather than parallelogram in 2D case. Since a purely Lagrangian method is always associated with mesh entangling, in order to avoid computational collapsing in multidimensional flow simulation, the mesh velocity is constructed by considering both fluid velocity (Lagrangian methodology) and diffusive velocity (Regenerating Eulerian mesh function). Therefore, we obtain a generalized Arbitrary-Lagrangian–Eulerian (ALE) method by properly designing a mesh velocity instead of re-generating a new mesh after distortion. As a result, the remapping step to interpolate flow variables from old mesh to new mesh is avoided. The current method provides a general framework, which can be considered as a remapping-free ALE-type method. Since there is great freedom in choosing mesh velocity, in order to improve the accuracy and robustness of the method, the adaptive moving mesh method [H.Z. Tang, T. Tang, Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515] can be also used to construct a mesh velocity to concentrate mesh to regions with high flow gradients.

1. Introduction

In fluid dynamics computations, two mesh systems are usually used: the Eulerian mesh to describe fluid motion at fixed location, and the Lagrangian mesh to follow fluid elements with fluid velocity. Considerable progress has been made over the past decades in both approaches [2,9,33,39].

In most numerical simulations, a computational domain including the boundary is fixed. In this case, the Eulerian method can work very well in many applications. However, for unsteady flow calculations, moving boundaries or interfaces often appear, such as the fluttering airfoils and multiphase flows. In these cases, the development of reliable methods for dynamically deforming computational domains is required [30,32].

For a single phase flow with fixed computational boundary, in order to improve numerical accuracy, the development of moving mesh method is still beneficial. Currently, there are many moving mesh methods in the literature. One example is the static mesh movement method [34,35,4,31,1], where after an Eulerian step, a new mesh is generated according to certain requirement. For example, according to a monitor function, the new mesh is concentrated to regions with high flow

* Corresponding author. Tel.: +86 10 62057509x2968; fax: +86 10 62057289.
E-mail addresses: gxni@iapcm.ac.cn (G. Ni), jiang@iapcm.ac.cn (S. Jiang), makxu@ust.hk (K. Xu).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.01.013

gradients. Then, all flow variables are interpolated from the old to the new mesh at an instant of time. Certainly, in order to keep the conservation, the flow update in the new mesh can be obtained using an equivalent flux function as the mesh moves across a static flow distribution. Another moving mesh method is the dynamical one, such as the Lagrangian method [25,37], where the fluid variables are updated inside each moving control volume within a time step. The dynamic moving mesh method is mostly used to track fluid interfaces with topological change and resolve interface physics. In order to avoid mesh distortion and tangling in the Lagrangian method, a widely used Arbitrary-Lagrangian–Eulerian (ALE) technique was developed [10,23]. In the ALE method, the numerical algorithm varies from a purely Lagrangian method to a purely Eulerian one through remapping so as to fix a distorted mesh. Due to the use of smoothing techniques in the remapping procedure, a regular mesh topology can be maintained to keep the accuracy of the numerical solution. The ALE method consists of three phases. The explicit Lagrangian phase, the rezoning phase, and the remapping phase. The rezoning phase was often carried out by employing a process of grid generation, for which it is a critical step to keep the geometric quality of the mesh [38,6]. The remapping phase is also important in the ALE method to interpolate flow variables between meshes [19,22]. The philosophy in the above approach is to remedy the mesh after distortion. Here, in our proposed scheme we are going to avoid distorted mesh starting from the first place.

Recently, a successful moving mesh method for the Euler equations has been developed by Hui et al. for the purpose of capturing the slip line crisply [13,14]. In this unified coordinate method, with a prescribed grid velocity, the inviscid flow equations are transformed into a conservative form in the computational domain (λ, ξ, η) along with the geometric conservation laws which control the mesh movement. In the unified coordinate method, the fluid equations and geometric evolution equations are coupled into an enlarged system, which is different from the transformed fluid equations alone [11]. Furthermore, due to the coupling of the fluid and geometric equations, for the first time a multidimensional Lagrangian gas dynamic system has been written in a conservative form. Theoretically, it was realized that the multidimensional Lagrangian system is only weakly hyperbolic. Numerically, for the unified coordinate method the fluid and geometric variables are updated simultaneously. In order to overcome the disadvantage in the Lagrangian method, the unified coordinate system adopts a grid velocity, which is set to be equal to $h\mathbf{q}$, where $\mathbf{q} = (U, V)$ is the fluid velocity and h is a parameter which is to be determined by other conditions, such as the mesh alignment with the slip surface or keeping grid angle fixed during the mesh movement. Therefore, the grid velocity can be changed locally according to the value of h. In a recent paper [15], the grid velocity has been further generalized to (hU, kV), where h and k are two parameters to be determined. An achievement of the unified coordinate method is that the numerical diffusion across the slip line can be avoided or reduced to a minimum level by enforcing the mesh without crossing the slip surface. However, in the complicated flow movement, in order to avoid the severe mesh distortion, the constraints, such as mesh orthogonality, have to be used. As a result, in most cases, this constraint automatically gives zero mesh velocity. Also, due to the coupling between the fluid and the geometrical equations, the mesh topology is implicitly required to be parallelogram in 2D and parallelepiped in 3D cases, which is hardly satisfied for complicated flow problems. So, in order to develop a generalized moving mesh method for any quadrilateral mesh in 2D case, instead of starting with the differential equations, it is necessary to begin with the integral equations directly.

Based on the unified coordinate transformation, a gas-kinetic scheme has been developed recently [17,18]. The corresponding Navier–Stokes solutions in the moving system have been obtained. In the gas-kinetic representation, all macroscopic flow variables are the moments of a single gas distribution function and the inviscid and viscous effects can be simply recovered through particle transport and collision processes [41], which is especially beneficial for the accurate capturing of supersonic viscous and heat conducting flows. The unified gas-kinetic method includes the following ingredients. First, the gas-kinetic BGK equation is transformed from the physical space (t, x, y) to a computational space (λ, ξ, η). Then, the corresponding Euler and Navier–Stokes equations in the computational domain are obtained using the Chapman–Enskog expansion. Second, a corresponding flux function across the computational interface in (λ, ξ, η) space is constructed. The advantage of the method, as for the unified coordinate system [14], is that an exact conservative mathematical formulation for both fluid and geometrical variables is used and kept in the numerical scheme. Since the gas-kinetic scheme solves the viscous governing equations intrinsically, it is hard to resolve the Eulerian solution, such as the sharp slip discontinuity in the unified coordinate system.

In this paper, we are going to construct a kinetic scheme on moving mesh for both Euler and Navier–Stokes equations. In order to construct a scheme under any irregular mesh configuration, instead of using the unified coordinate transformation, we are going to start with the integral equations of the conservation laws directly, and solve it by an finite volume scheme with arbitrary mesh velocity. Since the numerical flux can be obtained directly across a moving interface with arbitrary moving mesh velocity by solving the gas-kinetic equation, the current moving mesh BGK (MMBGK) method can incorporate the ALE ingredients into a properly designed mesh velocity so as to avoid severe mesh distortion. At the same time, the flow physics can be mostly captured. The determination of the mesh velocity is an important part of the current method. The methodology is that besides using the fluid velocity, such as the Lagrangian method, a diffusive or dissipative mesh velocity has to be considered to keep mesh regularity. Since we believe that an optimal mesh velocity depends on individual flow problem, under the general methodology the specific mesh velocity for different problems will be studied in this paper. For example, for one-dimensional numerical examples an elliptic solver with a monitor function is used for the new mesh generation, and subsequently the mesh velocity is determined by comparing the difference between the old and new meshes. For the two-dimensional cases, a diffusive mesh velocity is designed and added into the Lagrangian formulation. Therefore, the new method can not only keep the mesh quality, but also get accurate physical solutions. The merits of

the current method is its unification of the ALE method into a single moving mesh method framework. In comparison with the existing moving mesh methods and ALE methods, the remapping step for the flow variable interpolation between meshes is absent, this subsequently avoids the numerical errors induced by the remapping process. Usually, the accuracy could not be guaranteed in the remapping step when the mesh is irregularly distributed. The current method can be also considered as a unification of static and dynamic moving mesh methods. The current method is based on the integral equations directly. The decoupling between the fluid equations and geometrical equations makes the method more general and flexible for flow simulations with a defined mesh moving velocity.

This paper is organized as follows: In Section 2, we construct the finite volume method based on the integral equations and the gas-kinetic flux evaluation on a moving mesh. Section 3 is dedicated to the determination of the mesh velocity. Numerical experiments are given in Section 3. The last section is the conclusion.

2. Kinetic scheme on moving meshes

In this section, based on the integral equations with moving boundary, we will first construct the corresponding finite volume scheme. Then, the gas-kinetic scheme will be used in the evaluation of numerical fluxes across a moving cell interface.

2.1. Semi-discrete scheme

The conservative system for the mass, momentum and energy can be written in a unified differential form for both the Euler and the Navier–Stokes equations [20]. An integral form over a moving control volume $\Omega(t)$ with an arbitrary boundary velocity $\mathbf{U}_g$ can be written in the following form (see [10]),

$$\begin{cases} \frac{d}{dt}\int_{\Omega(t)} \rho\, dv = -\int_{S(t)} \rho(\mathbf{U}-\mathbf{U}_g)\cdot\mathbf{n}\, ds, \\ \frac{d}{dt}\int_{\Omega(t)} \rho\mathbf{U}\, dv = -\int_{S(t)} \rho\mathbf{U}(\mathbf{U}-\mathbf{U}_g)\cdot\mathbf{n}\, ds, + \int_{S(t)} \Pi\cdot\mathbf{n}\, ds, \\ \frac{d}{dt}\int_{\Omega(t)} \rho E\, dv = -\int_{S(t)} \rho E(\mathbf{U}-\mathbf{U}_g)\cdot\mathbf{n}\, ds + \int_{S(t)} \Pi\cdot\mathbf{U}\cdot\mathbf{n}\, ds - \int_{S(t)} \mathbf{q}\cdot\mathbf{n}\, ds. \end{cases} \tag{2.1}$$

Here $\rho, \mathbf{U}, \rho E, \mathbf{\Pi}, \mathbf{q}$ are the density, velocity, total energy, stress, and heat flux of the fluid, respectively, i.e., $E = \frac{1}{2}\mathbf{U}^2 + e$ with e being the specific internal energy, $\mathbf{q} = -\kappa\nabla\Theta$ with Θ, κ being the temperature and the heat conductivity coefficient. The stress tensor is defined by

$$\Pi_{ij} = -\delta_{ij}P + \sigma_{ij}, \quad i,j = 1,2,$$

where σ_{ij} is the viscous stress, P is the pressure. $S(t)$ is the boundary of $\Omega(t)$, $\mathbf{n}$ is the outward unit normal vector of $S(t)$. Note that the above equations become the Eulerian form when $\mathbf{U}_g = 0$, and the Lagrangian form when $\mathbf{U}_g = \mathbf{U}$.

The boundary velocity $\mathbf{U}_g$ in (2.1) is a function of time and position. Theoretically, it can be arbitrary to control the volume $\Omega(t)$ evolution. The control volume $\Omega(t)$ is basically the computational cell.

Now, we consider a special polygon, such as a quadrilateral $\Omega = [ABCD]$ in Fig. 1. A piecewise constant boundary velocity on each side of a control volume, i.e., $\mathbf{U}_g^i (i = 1,\ldots,4)$, is assumed.

In order to evaluate the flux across the moving interface, it is important to consider the relative velocity of fluid to the mesh. Denote the relative velocity by

$$\widetilde{\mathbf{U}} = \mathbf{U} - \mathbf{U}_g,$$

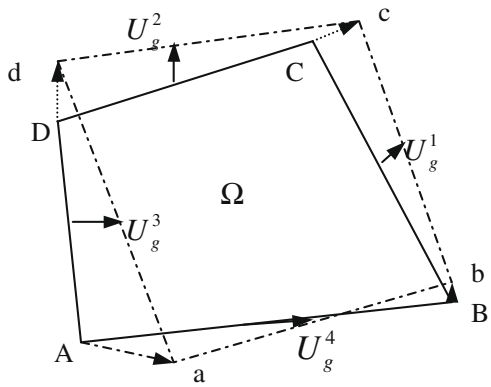


Fig. 1. Moving mesh. [ABCD] is the original mesh. [abcd] is the moved mesh with mesh velocity U_g^i in the corresponding edge.

and the corresponding value at edge e_i is given by

$$\widetilde{\mathbf{U}}^i = (\mathbf{U} - \mathbf{U}_g)|_{e_i},$$

where $e_1 = BC, e_2 = CD, e_3 = DA, e_4 = AB$ are the four edges of the quadrilateral $\Omega = [ABCD]$, see Fig. 1.

The semi-discretization of the integral equations (2.1) is the following. First, the continuity equation is discretized as

$$\frac{d}{dt}\int_{\Omega(t)} \rho\, dv = -\int_{S(t)} \rho(\mathbf{U} - \mathbf{U}_g)\cdot \mathbf{n}\, ds = -\int_{S(t)} \rho\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds = -\sum_{i=1}^{4}\int_{e_i} \rho\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds = \sum_{i=1}^{4}\mathcal{F}_{\rho,e_i}(t),$$

where

$$\mathcal{F}_{\rho,e_i}(t) = -\int_{e_i} \rho\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds$$

is the mass flux across the moving edge e_i. Since the stress tensor is a function of velocity gradient, with a constant mesh velocity $\mathbf{U}_g^i$ within a time step, the stress tensor remains unchanged in both stationary and moving mesh. Thus, the semi-discetization of the momentum equations is given by

$$\begin{aligned}\frac{d}{dt}\int_{\Omega(t)} \rho\mathbf{U}\, dv &= -\int_{S(t)} \rho\mathbf{U}(\mathbf{U} - \mathbf{U}_g)\cdot \mathbf{n}\, ds + \int_{S(t)} \Pi\cdot \mathbf{n}\, ds = -\int_{S(t)} \rho\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds + \int_{S(t)} \mathbf{\Pi}\cdot \mathbf{n}\, ds - \int_{S(t)} \rho\mathbf{U}_g\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds\\ &= \sum_{i=1}^{4}\{\mathcal{F}_{\rho\tilde{\mathbf{U}},e_i}(t) + \mathbf{U}_g^i\mathcal{F}_{\rho,e_i}(t)\},\end{aligned}$$

where

$$\mathcal{F}_{\widetilde{\rho\mathbf{U}},e_i}(t) = -\int_{S(t)} \rho\widetilde{\mathbf{U}}\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds + \int_{S(t)} \mathbf{\Pi}\cdot \mathbf{n}\, ds$$

is the momentum flux across the moving edge e_i. Similarly, the energy equation is semi-discretized by

$$\begin{aligned}\frac{d}{dt}\int_{\Omega(t)} \rho E\, dv &= -\int_{S(t)} \rho E(\mathbf{U} - \mathbf{U}_g)\cdot \mathbf{n}\, ds + \int_{S(t)} \mathbf{\Pi}\cdot \mathbf{U}\cdot \mathbf{n}\, ds - \int_{S(t)} \mathbf{q}\cdot \mathbf{n}\, ds\\ &= -\int_{S(t)} \rho\left(\frac{1}{2}\mathbf{U}^2 + e\right)\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds + \int_{S(t)} \mathbf{\Pi}\cdot \mathbf{U}\cdot \mathbf{n}\, ds - \int_{S(t)} \mathbf{q}\cdot \mathbf{n}\, ds\\ &= -\int_{S(t)} \rho\left(\frac{1}{2}\tilde{\mathbf{U}}^2 + e\right)\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds + \int_{S(t)} \Pi\cdot \widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds - \int_{S(t)} \mathbf{q}\cdot \mathbf{n}\, ds\\ &\quad + \int_{S(t)} \mathbf{\Pi}\cdot \mathbf{U}_g\cdot \mathbf{n}\, ds - \int_{S(t)} \rho\mathbf{U}_g\cdot \widetilde{\mathbf{U}}\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds - \frac{1}{2}\int_{S(t)} \rho\mathbf{U}_g^2\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds\\ &= \sum_{i=1}^{4}\left\{\mathcal{F}_{\widetilde{\rho E},e_i}(t) + \mathbf{U}_g^i\mathcal{F}_{\widetilde{\rho\mathbf{U}},e_i}(t) + \frac{1}{2}(\mathbf{U}_g^i)^2\mathcal{F}_{\rho,e_i}(t)\right\},\end{aligned}$$

where

$$\mathcal{F}_{\widetilde{\rho E},e_i}(t) = -\int_{S(t)} \rho\left(\frac{1}{2}\widetilde{\mathbf{U}}^2 + e\right)\widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds + \int_{S(t)} \mathbf{\Pi}\cdot \widetilde{\mathbf{U}}\cdot \mathbf{n}\, ds - \int_{S(t)} \mathbf{q}\cdot \mathbf{n}\, ds$$

is the energy flux across the moving edge e_i.

In summary, the semi-discrete form of integral equations (2.1) reads

$$\begin{cases}\frac{d}{dt}\int_{\Omega(t)} \rho\, dv = \sum_{i=1}^{4}\mathcal{F}_{\rho,e_i}(t),\\ \frac{d}{dt}\int_{\Omega(t)} \rho\mathbf{U}\, dv = \sum_{i=1}^{4}\{\mathcal{F}_{\widetilde{\rho\mathbf{U}},e_i}(t) + \mathbf{U}_g^i\mathcal{F}_{\rho,e_i}(t)\},\\ \frac{d}{dt}\int_{\Omega(t)} \rho E\, dv = \sum_{i=1}^{4}\left\{\mathcal{F}_{\widetilde{\rho E},e_i}(t) + \mathbf{U}_g^i\mathcal{F}_{\widetilde{\rho\mathbf{U}},e_i}(t) + \frac{1}{2}(\mathbf{U}_g^i)^2\mathcal{F}_{\rho,e_i}(t)\right\}.\end{cases} \tag{2.2}$$

In the numerical computation, an operator splitting method is used. At the cell interface, we have to project the velocity to normal and the tangential direction. More precisely, with the normal direction $\mathbf{n} = (\cos\alpha, \sin\alpha)$ of the edge e_i, and $\mathbf{U} = (U, V), \mathbf{U}_g = (U_g, V_g)$, the normal velocity component U' and tangential component V' are given by

$$\begin{cases}U' = U\cos\alpha + V\sin\alpha,\\ V' = -U\sin\alpha + V\cos\alpha,\end{cases}$$

and $\tilde{\mathbf{U}}' = (\tilde{U}', \tilde{V}')$. In this way, the above Eq. (2.2) can be written in component-wise form:

$$\begin{cases} \frac{d}{dt}\int_{\Omega(t)} \rho \, dv = \sum_{i=1}^{4} \mathcal{F}_{\rho,e_i}(t), \\ \frac{d}{dt}\int_{\Omega(t)} \rho U \, dv = \sum_{i=1}^{4} \{\mathcal{F}_{\rho\tilde{U}',e_i}(t)\sin\alpha + \mathcal{F}_{\rho\tilde{V}',e_i}(t)\cos\alpha + U_g^i \mathcal{F}_{\rho,e_i}(t)\}, \\ \frac{d}{dt}\int_{\Omega(t)} \rho V \, dv = \sum_{i=1}^{4} \{-\mathcal{F}_{\rho\tilde{U}',e_i}(t)\cos\alpha + \mathcal{F}_{\rho\tilde{V}',e_i}(t)\sin\alpha + V_g^i \mathcal{F}_{\rho,e_i}(t)\}, \\ \frac{d}{dt}\int_{\Omega(t)} \rho E \, dv = \sum_{i=1}^{4} \Big\{ \mathcal{F}_{\rho\widetilde{E},e_i}(t) + (U_g^i \sin\alpha - V_g^i \cos\alpha)\mathcal{F}_{\rho\tilde{U}',e_i}(t) + (U_g^i \cos\alpha + V_g^i \sin\alpha)\mathcal{F}_{\rho\tilde{V}',e_i}(t) \\ \qquad\qquad + \frac{1}{2}\left((U_g^i)^2 + (V_g^i)^2\right)\mathcal{F}_{\rho,e_i}(t)\Big\}, \end{cases} \tag{2.3}$$

where the numerical fluxes on the right-hand side of (2.3) are along the normal direction of the moving cell interface e_i. We will show in the next section how to evaluate the numerical fluxes by using a gas-kinetic scheme. The above equations represent precisely a finite volume scheme with moving cell interfaces.

2.2. Gas-kinetic flux function: a generalized Riemann solution for the Navier–Stokes equations

The time evolution of the conservative flow variables in (2.3) depends on the flux evaluation on a moving reference of frame with grid velocity $\mathbf{U}_g^i$. Theoretically, any flux function for the Navier–Stokes equations can be used here, such as a Riemann solver for the inviscid fluxes and the central difference for the viscous part. In this paper, we shall use the gas-kinetic scheme for the flux evaluation. The main reasons using the gas-kinetic scheme are that the gas-kinetic flux function is time-accurate, therefore, the semi-discrete form (2.3) can be integrated in time explicitly. At the same time, the inviscid and viscous fluxes are obtained simultaneously through the solution of a single gas distribution function under the initial condition of a generalized Riemann problem.

For compressible flow simulations, a finite volume gas-kinetic BGK scheme has been developed and applied to many physical and engineering problems [5,41,16,26]. Similar to many other finite volume methods, the crucial step in the construction of a gas-kinetic scheme is the flux evaluation at a cell interface. The distinguishable feature of the gas-kinetic BGK scheme is that a Navier–Stokes flux is given directly from the MUSCL-type reconstructed initial data [36], where both flow variable gradients on the left and right sides of a cell interface participate in the construction of the time-dependent solution.

In this section, we will construct the numerical fluxes in (2.3) on moving meshes by using the BGK model. The difference of our moving mesh kinetic scheme from the scheme in an Eulerian reference of frame is that all velocities presented in this subsection are the relative velocity. The simplicity of the gas-kinetic scheme for a moving mesh system is due to the fact that the particle transport is always on a straight line in any inertia reference of frame.

The BGK model in two dimensions can be written as (see [3,40,41])

$$f_t + uf_x + vf_y = \frac{g - f}{\tau}, \tag{2.4}$$

where f is the gas distribution function and g is the equilibrium state approached by f, (u, v) is the particle velocity. Both f and g are functions of x, y, t, u, v and the internal variable ξ. The particle collision time τ is related to the viscosity coefficient.

Generally, the equilibrium state is a Maxwellian distribution

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{\frac{K+2}{2}} e^{-\lambda[(u-\tilde{U}')^2 + (v-\tilde{V}')^2 + \xi^2]}, \tag{2.5}$$

where $\tilde{\mathbf{U}}' = (\tilde{U}', \tilde{V}')$ is the relative velocity as before, $\lambda = m/(2kT)$ with m, k and T being the molecular mass, the Boltzmann constant and the temperature, respectively. The total number of degree of freedom K in ξ is equal to $(4 - 2\gamma)/(\gamma - 1)$ with γ being the specific heat ratio, and ξ^2 denotes $\xi^2 = \xi_1^2 + \xi_2^2 + \cdots + \xi_K^2$.

The relation between the mass ρ, the momentum $(\rho\tilde{U}', \rho\tilde{V}')$, the energy $\rho\widetilde{E}$ and the distribution function f is given by

$$(\rho, \rho\tilde{U}', \rho\tilde{V}', \rho\widetilde{E})^T = \int \Psi f \, d\Xi,$$

where

$$\Psi = (\psi_1, \psi_2, \psi_3, \psi_4)^T = \left(1, u, v, \frac{1}{2}(u^2 + v^2 + \xi^2)\right)^T,$$

and $d\Xi = du\,dv\,d\xi$ is the volume element in the phase space.

Since the mass, momentum, and energy are conservative quantities during particle collisions, f and g satisfy the conservation constraints

$$\int (g-f)\psi_\alpha d\Xi = 0, \quad \alpha = 1,2,3,4,$$

at any point in space and time.

For a local equilibrium state with $f = g$, taking the moments of Ψ to the Eq. (2.4), we find that

$$\int (g_t + ug_x + vg_y)\Psi\, d\Xi = 0, \tag{2.6}$$

from which the corresponding Euler equations with $e = \frac{K+2}{4\lambda}$ and $P = \rho/(2\lambda)$ can be obtained.

On the other hand, to the first order of τ, the Chapman–Enskog expansion gives $f = g - \tau(g_t + ug_x + vg_y)$. Taking the moments of Ψ to the BGK Eq. (2.4) with this f, we get

$$\int (g_t + ug_x + vg_y)\Psi d\Xi = \tau \int (g_{tt} + 2ug_{xt} + u^2 g_{xx} + 2vg_{yt} + 2uvg_{xy} + v^2 g_{yy})\Psi\, d\Xi, \tag{2.7}$$

from which the compressible Navier–Stokes equations can be derived.

Therefore, from (2.6) and (2.7) we see that as soon as the solution of (2.4) is known, the inviscid and viscous fluxes can be obtained by taking appropriate moments of Ψ to f. This is in fact the basic idea in the construction of numerical fluxes of gas-kinetic schemes.

The general solution f of (2.4) at any point (x, y) and time t is given by

$$f(x,y,t,u,v) = \frac{1}{\tau}\int_0^t g(x',y',t',u,v)e^{-(t-t')/\tau}\,dt' + e^{-t/\tau} f_0(x-ut, y-vt), \tag{2.8}$$

where $x' = x - u(t-t')$ and $y' = y - v(t-t')$.

To construct a second order scheme, we only need to use a MUSCL-type conservative flow variable distributions on the left- and right-hand sides of a cell interface, and to obtain the corresponding g, f_0 in (2.8) at the beginning of each time step. The details are omitted here. Once f_0 and g in (2.8) are determined, the fluxes in (2.3) across a cell interface can be obtained by taking moments of Ψ to the solution f given in (2.8):

$$\mathcal{F}_{\Psi,e_i}(t) = (\mathcal{F}_{\rho,e_i}, \mathcal{F}_{\rho\tilde{U}',e_i}, \mathcal{F}_{\rho\tilde{V}',e_i}, \mathcal{F}_{\rho\tilde{E},e_i})^T(t) = \int \Psi f(0,0,t,u,v)\, d\Xi, \tag{2.9}$$

where $\Psi = (\psi_1, \psi_2, \psi_3, \psi_4)^T$ are the moments defined before. The particle collision time τ in the above gas distribution function is determined by the viscosity coefficient. For example, at each cell interface $\tau = \mu/P$ is used, where μ is the local dynamical viscosity coefficient which may depend on local density and temperature.

2.3. Fully-discrete scheme

The numerical flux in (2.9) is time-accurate [41]. The integration over a time step from t^n to t^{n+1} is to yield

$$F_{\Psi,e_i} = \int_{t^n}^{t^{n+1}} \mathcal{F}_{\Psi,e_i}(t)\, dt.$$

Thus, integrating (2.3) with respect to time and using the above identity, we obtain

$$\begin{cases}
\int_{\Omega'} \rho\, dv = \int_\Omega \rho\, dv + \sum_{i=1}^4 F_{\rho,e_i}, \\
\int_{\Omega'} \rho U\, dv = \int_\Omega \rho U\, dv + \sum_{i=1}^4 \{F_{\rho\tilde{U}',e_i}\sin\alpha + F_{\rho\tilde{V}',e_i}\cos\alpha + U_g^i F_{\rho,e_i}\}, \\
\int_{\Omega'} \rho V\, dv = \int_\Omega \rho V\, dv + \sum_{i=1}^4 \{-F_{\rho\tilde{U}',e_i}\cos\alpha + F_{\rho\tilde{V}',e_i}\sin\alpha + V_g^i F_{\rho,e_i}\}, \\
\int_{\Omega'} \rho E\, dv = \int_\Omega \rho E\, dv + \sum_{i=1}^4 \left\{F_{\rho\tilde{E},e_i} + (U_g^i\sin\alpha - V_g^i\cos\alpha)F_{\rho\tilde{U}',e_i} + (U_g^i\cos\alpha + V_g^i\sin\alpha)F_{\rho\tilde{V}',e_i} + \tfrac{1}{2}((U_g^i)^2 + (V_g^i)^2)F_{\rho,e_i}\right\},
\end{cases} \tag{2.10}$$

where $\Omega' = \Omega(t^{n+1}) = [abcd]$ and $\Omega = \Omega(t^n) = [ABCD]$, see Fig. 1. If we denote $\overline{A} = \frac{1}{|\Omega|}\int_\Omega A(x)\, dx$, we get the fully discrete form of (2.10):

$$\begin{cases}
\overline{\rho^{n+1}} = \left\{\overline{\rho^n}|\Omega| + \sum_{i=1}^4 F_{\rho,e_i}\right\}\Big/|\Omega'|, \\
\overline{\rho U^{n+1}} = \left\{\overline{\rho U^n}|\Omega| + \sum_{i=1}^4 [F_{\rho\tilde{U}',e_i}\sin\alpha + F_{\rho\tilde{V}',e_i}\cos\alpha + U_g^i F_{\rho,e_i}]\right\}\Big/|\Omega'|, \\
\overline{\rho V^{n+1}} = \left\{\overline{\rho V^n}|\Omega| + \sum_{i=1}^4 [-F_{\rho\tilde{U}',e_i}\cos\alpha + F_{\rho\tilde{V}',e_i}\sin\alpha + V_g^i F_{\rho,e_i}]\right\}\Big/|\Omega'|, \\
\overline{\rho E^{n+1}} = \left\{\overline{\rho E^n}|\Omega| + \sum_{i=1}^4 \left[F_{\rho\tilde{E},e_i} + (U_g^i\sin\alpha - V_g^i\cos\alpha)F_{\rho\tilde{U}',e_i} + (U_g^i\cos\alpha + V_g^i\sin\alpha)F_{\rho\tilde{V}',e_i} + \tfrac{1}{2}((U_g^i)^2 + (V_g^i)^2)F_{\rho,e_i}\right]\right\}\Big/|\Omega'|.
\end{cases} \tag{2.11}$$

The conservative flow variables can be updated once the numerical fluxes are evaluated on a moving cell interface. In the above formulation, in the 2D case we basically don't require a quadrangular mesh. Actually, Ω can be any shape with any number of linear sections to compose of the boundary of the control volume.

3. Mesh velocity determination and numerical experiments

In this section, we will discuss how to construct the mesh velocity $\mathbf{U}_g$. In the multidimensional case, the method of constructing $\mathbf{U}_g$ here is functionally equivalent to unify the three steps in ALE method by considering the mesh regularity in the mesh velocity determination. In the one-dimensional case, based on the use of a homogeneous elliptic solver [34], a new mesh will be generated at the beginning of each time step. Then, the mesh velocity will be determined by comparing the location of the new mesh with that of the old mesh. In the two-dimensional case, if we simply set $\mathbf{U}_g$ to be equal to the fluid velocity $\mathbf{U}$, a Lagrangian method is obtained. In this case, it is very difficult to keep the mesh regularity, and the mesh will get distorted and entangled under large deformation. Therefore, in two dimensions, besides using the fluid velocity $\mathbf{U}$, we will add a suitable diffusive mesh velocity to smooth out the possibly distorted meshes in the first place. Since there are infinite number of ways to distort the mesh numerically, most of them cannot be properly described on the partial differential equation level. So, it is more reasonable to define a diffusive mesh velocity directly in the discretized level by modifying local mesh movement to prevent it from collapsing to neighboring ones.

In the following, we will discuss the one- and two-dimensional cases and give numerical examples, respectively.

3.1. Mesh velocity $\mathbf{U}_g$ determined through an elliptic solver

Certainly, many methods can be used to determine the mesh velocity $\mathbf{U}_g$. Here the mesh velocity $\mathbf{U}_g$ is determined by solving an elliptic solver [12,34], which is equivalent to optimizing some functionals in the whole computational domain [38].

For the one-dimensional case, at $t = t_n$ the grid location x_i^n and the flow variables are given. Assume that the new location x_i^{n+1} of the new mesh has been already obtained, then the mesh velocity U_g for each node is defined by

$$U_{g,i} = (x_i^{n+1} - x_i^n)/\Delta t.$$

Thus, the key point in the construction of U_g is the determination of the position x_i^{n+1} using the known conditions at $t = t_n$. Generally speaking, in order to generate a new mesh, two aspects have to be considered. First, the new mesh should be smooth. Second, the new mesh will be automatically redistributed and concentrated in the region with high gradients of the flow variables in order to improve the computational accuracy. These two requirements can be satisfied using the approach of mesh movement, which is based on the variational approach and described below.

Let x and ξ denote the physical and computational coordinates. A coordinate mapping from the computational domain Ω_c to the physical domain Ω_p is given by $x = x(\xi)$, and the inverse map becomes $\xi = \xi(x)$. The specific map for the mesh is obtained by minimizing a functional with the following form

$$\widetilde{E}[\xi] = \frac{1}{2}\int_{\Omega_c} (\omega x_\xi)^2 \, d\xi, \tag{3.1}$$

where ω is a monitor function which controls the distribution of the mesh. The Euler–Lagrange equations can be obtained from (3.1)

$$(\omega x_\xi)_\xi = 0. \tag{3.2}$$

Therefore, the mesh distribution in the physical space can be directly generated by solving Eq. (3.2). In the above formula, ω is a function of the flow variables, such as the density, velocity, pressure, or their gradients. For example, one can take

$$\omega = \sqrt{1 + \alpha f^2 + \beta f_x^2}, \tag{3.3}$$

where f is some variable, and α and β are nonnegative parameters, which are problem-dependent.

A second order central difference scheme can be used to discretize the mesh generation Eq. (3.2) and the resulting algebraic equations can be solved numerically by iterative methods, such as the Gauss–Seidel iteration. If the monitor function ω is assumed to be constant, i.e., $\omega = 1$, the Euler–Lagrange equations (3.2) become the well-known Laplace's equation, and consequently, a smooth mesh can be generated.

The extension of the above method to 2D case is straightforward. With the newly generated mesh, the mesh velocity can be considered as an optimal one if the target is to minimizing or maximizing certain functionals.

We now present some numerical examples. In all cases, the time step is determined by

$$\Delta t = \mathrm{CFL} \min_j \sqrt{|\Omega_j|}/(\max|u| + c),$$

where CFL is the CFL number and c is the sound speed. In our numerical examples, the CFL number varies between $0.25 \sim 0.45$.

Example 1 (*Lax's shock tube problem*). This test is a Riemann problem with initial data

$$\begin{cases} (\rho, U, P) = (0.445, 0.698, 3.528), & x \in (0.0, 5.0), \\ (\rho, U, P) = (0.5, 0, 0.571), & x \in (5.0, 10.0). \end{cases}$$

It was proposed by Lax in his early paper, and has been extensively studied by many authors. The computational domain is $x \in [0, 10]$, where 200 mesh points are used to cover the domain. We choose $f = \rho$ in the monitor function (3.3) with $\alpha = 1.0$ and $\beta = 0.1$. The computed results by the current scheme (the MMBGK scheme) are shown in Fig. 2, where the solid line is the exact solution. From Fig. 2 we see that the effect of mesh movement on the solution accuracy is obvious due to the concentration of mesh points to the region of rapid changes in the flow variables. The solution obtained using the current second order MMBGK is very close to that obtained using the third order DGBGK scheme from [27].

Example 2 (*Entropy wave*). This is an example of shock interaction with an entropy wave. A Mach 3 moving shock interacts with the sine wave, i.e.,

$$\begin{cases} (\rho, U, P) = (3.857143, 2.629369, 10, 333333), & x \in (0.0, 1.0), \\ (\rho, U, P) = (1 + \epsilon \sin(5x), 0, 1), & x \in (1.0, 10.0), \end{cases}$$

where $\epsilon = 0.2$. The reference ("exact") solution for this problem is computed by the fifth-order finite difference WENO scheme with 2000 grid points [28]. In the current MMBGK calculation, the whole domain $x \in [0, 10]$ is covered with 200 mesh points. We take $f = P$ in the monitor function with $\alpha = 1.0$ and $\beta = 1.0$. The computed density distribution is shown in Fig. 3. In comparison with the reference solution, it is obvious that high accurate solution has been obtained. The accuracy of the current second order MMBGK method is very close to that of a third order DGBGK scheme [27] with a third order WENO limiter [28].

Example 3 (*Blast wave interaction*). In this example, we test the interaction of blast waves, see [39]. The initial data read

$$\begin{cases} (\rho, U, P) = (1, 0, 1000), & x \in (0.0, 1.0), \\ (\rho, U, P) = (1, 0, 0.01), & x \in (1.0, 9.0), \\ (\rho, U, P) = (1, 0, 100), & x \in (9.0, 10.0). \end{cases}$$

The reference ("exact") solution is computed again by the fifth-order finite difference WENO scheme with 2000 cells [28]. In the current calculation, 400 grid points are used in the domain and the reflecting boundary conditions are applied at left and right boundaries. In our computation we take $f = \rho$ in the monitor function with $\alpha = 1.0$ and $\beta = 1.0$. The computed density contours are shown in Fig. 4. The accuracy of the solutions from the current scheme is very close to that obtained using the third order DGBGK method [27], and is much better than that obtained using a second order Lagrangian method [7].

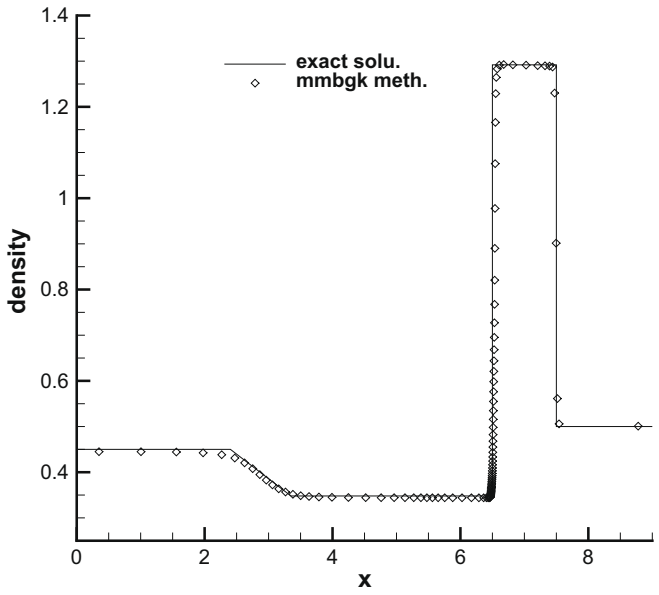


Fig. 2. Lax problem. Density distributions at time $t = 1.0$ with 200 mesh points. Circles are the numerical results of the MMBGK scheme and the solid line is the exact solution.

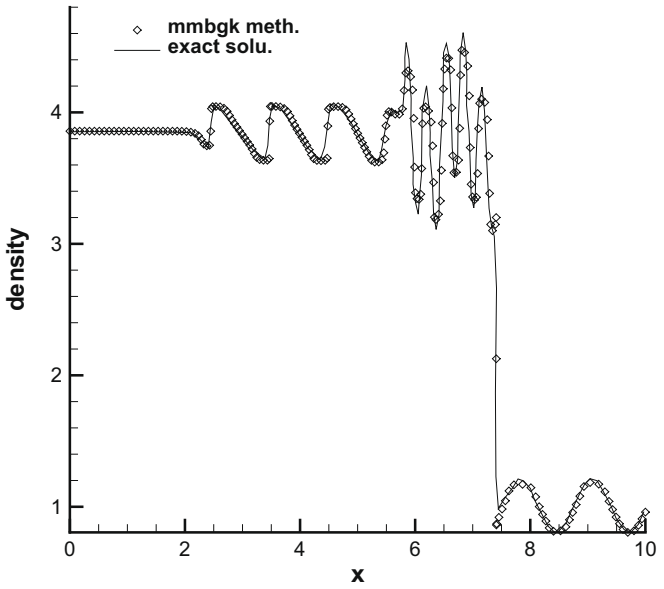


Fig. 3. Entropy waves. Density distributions at time $t = 1.8$ with 200 mesh points. Circles are the numerical results of the MMBGK method and the solid line is the "exact" solution.

Example 4 (*2D Riemann problem*). We test a two-dimensional Riemann problem which has been studied by many authors [21,34]. In this test, the determination of the mesh velocity is the same as that for the one-dimensional case. The static moving mesh method generates a mesh to move mesh points to the flow regions with high gradients.

The initial data for the 2D Riemann problem are

$$(\rho, U, V, P) = \begin{cases} (1.1, 0.0, 0.0, 1.1), & x > 0.5,\ y > 0.5, \\ (0.5065, 0.8939, 0.0, 0.35), & x < 0.5,\ y > 0.5, \\ (1.1, 0.8939, 0.8939, 1.1), & x > 0.5,\ y < 0.5, \\ (0.5065, 0.0, 0.8939, 0.35), & x > 0.5,\ y < 0.5. \end{cases}$$

We carry out the computation in the domain $(x, y) \in [0, 1] \times [0, 1]$ by using both 50×50 and 100×100 grid points. The mesh and density distributions at time $t = 0.25$ are shown in Figs. 5–7. It is easy to observe that the current scheme sharply resolves shock waves and the solutions are convergent.

3.2. Mesh velocity $\mathbf{U}_g$ determined on the discretized level

In the following examples, we are going to consider to add diffusive mesh velocity in the Lagrangian formulation to remedy possible mesh distortion. Since mesh distortion can be hardly described mathematically on the differential equation level, a direct implementation of numerical techniques on the discretized level to prevent mesh distortion is preferred.

Example 5 (*Saltzman's piston problem*). This problem tests the ability of numerical methods to capture the shock propagation through a systematically distorted mesh, which has become a benchmark test case [8,7]. The set-up of the problem is the following. In the computational domain $(x, y) \in [0, 1] \times [0, 0.1]$, a mesh with 100×10 mesh points is used with the nodal x coordinates

$$x_{ij} = (i-1)\Delta x + (11-j)\sin\left(\frac{\pi(i-1)}{100}\right)\Delta y,$$

and a uniform nodal y coordinates with $\Delta x = \Delta y = 0.01$.

An ideal monatomic gas with the specific heat ratio $\gamma = 5/3$ is filled in the box. The left boundary is a movable piston. Starting from $t = 0$, the piston is pushing into the box with a constant velocity 1.0. As a consequence, a strong shock wave is generated from the left end. On the up and down boundaries, a simple extrapolation is used as the boundary condition. This problem has the exact solution. At time $t = 0.6$, the shock is expected to be located at $x = 0.8$, and the post shock solutions are $\rho = 0.6, U = 1.0$ (the velocity in the x-direction), and $P = 1.333$.

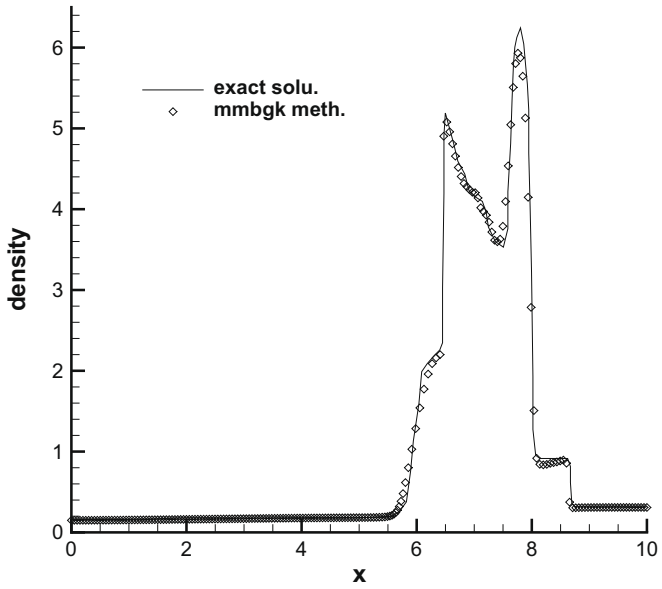


Fig. 4. Blast waves. Density distributions at time $t = 0.038$ with 400 mesh points. Circles are the numerical results of the MMBGK method and the solid line is the "exact" solution.

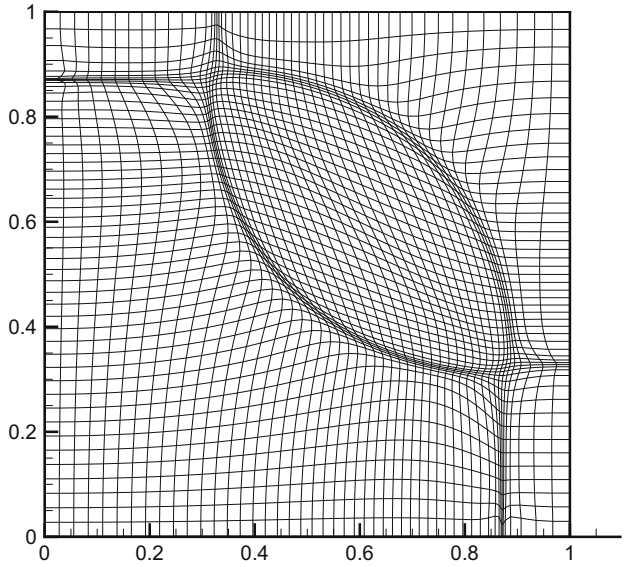


Fig. 5. 2D Riemann problem. Mesh distribution of the MMBGK method with 50×50 mesh points at $t = 0.25$.

Usually this problem is studied by Lagrangian method. Since the left side is pushed by a piston, the mesh on the left boundary has to move together with the piston. However, a purely Lagrangian method will cause mesh deformation, so an ALE method becomes a proper choice for this kind of problem. Here we are going to follow the Lagrangian method as much as possible. In order to avoid the mesh distortion, all ALE steps are merged into a single moving mesh method, where the mesh moving velocity $\mathbf{U}_g = (U_g, V_g)$ is controlled by both fluid velocity $\mathbf{U} = (U, V)$ and a newly defined diffusive velocity $\mathbf{U}_d = (U_d, V_d)$, i.e., $\mathbf{U}_g = \mathbf{U} + \mathbf{U}_d$. For this problem, since the piston is moving from left to right in the x-direction, the grid velocity in the y-direction is set to be

$$V_g = V + V_d = 0.$$

So, we only need to define the grid velocity in the x-direction U_g with the inclusion of the diffusive one U_d.

Let $x(i,j)$ denote the center of the cell Ω_{ij}, we define the relative distance differences among cell centers by

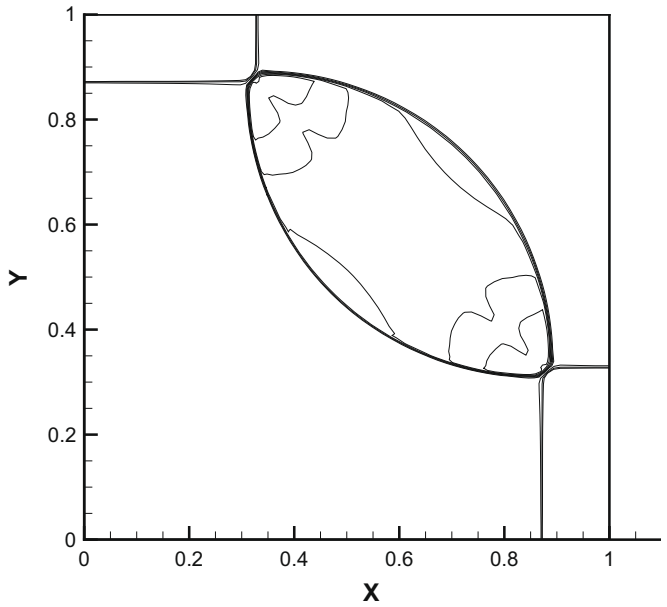


Fig. 6. 2D Riemann problem. Density distribution of the MMBGK method with 50×50 mesh points at $t = 0.25$. 14 equally spaced contour lines are used for the density plots between 0.5 and 2.5.

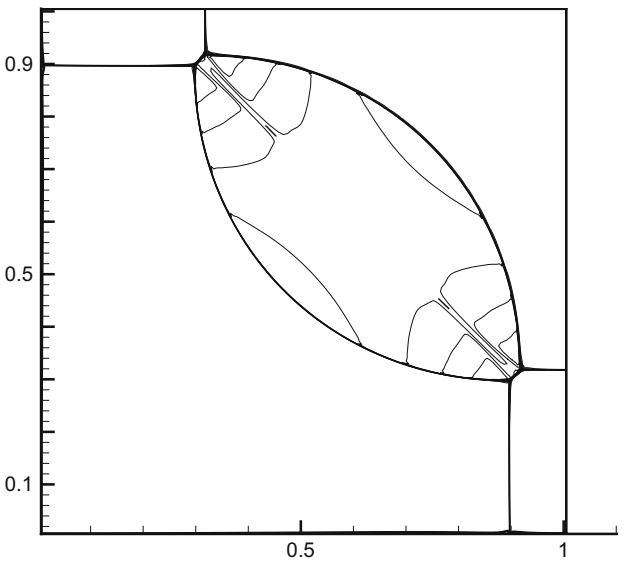


Fig. 7. 2D Riemann problem. Density distribution of the MMBGK method with 100×100 mesh points at $t = 0.25$. 14 equally spaced contour lines are used for density plots between 0.5 and 2.5.

$$\begin{cases} DL = x(i,j) - x(i-1,j), \\ DR = x(i+1,j) - x(i,j). \end{cases}$$

Then, a diffusive velocity U_d can be set to be equal to

$$U_d = \max\{DR - DL, 0.0\}/(DR + DL),$$

to push the mesh away from each other on the discretized level. Thus, for each point $x(i,j)$, the mesh velocity in the x-direction is given by

$$U_g = U + \mu U_d,$$

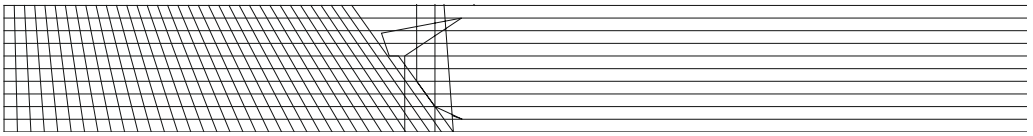
where U is the x-component of the fluid velocity and μ is the diffusive coefficient, which takes a fixed value $\mu = 1.0$ in the numerical calculation.

Based on the above choice $\mathbf{U}_g$, the mesh velocity at a cell interface is the average of $\mathbf{U}_g$ in its neighboring cell centers. The diffusive term $\mathbf{U}_d$ plays a role of preserving good quality of the mesh, which is similar to the remapping function of the ALE method.

The numerical results using the MMBGK scheme are shown in Figs. 8–13. The initial mesh is presented in Fig. 8, while Fig. 9 gives the mesh at time $t = 0.6$. The density contours are shown in Fig. 10. The density, pressure, and velocity profiles are shown in Figs. 11–13 along the line $y = 0.05$. The current method gives very accurate numerical solution in comparison with the exact solution and other numerical results [8,7].

Example 6 (*Sedov's blast wave problem*). This is a two-dimensional explosion problem to model blast wave from an energy-deposited singular point. Initially, a uniform medium with zero pressure is distributed inside the domain. A fixed amount of energy is deposited initially at the origin. As time increases, a blast wave expands away from the origin. Because the initial pressure outside singular point is zero, the shock strength associated with the blast wave is infinite, and a similarity solution for the post flow distribution can be obtained. The solution was given by Sedov in 1959 [29]. The exact solution is particularly useful for testing the accuracy of multidimensional computational schemes. It has become a standard benchmark problem for the Lagrangian method.

Here we run a two-dimensional calculation using our MMBGK method. The computational domain is $(x, y) \in [0, 1.1] \times [0, 1.1]$, an uniform mesh with 44×44 grid points is used. Initially, one sets the density $\rho = 1.0$, the velocity 0.0, and the internal energy $e = 0$ for all cells except the cell at origin with an amount of internal energy $e = 393.64$. At time $t = 1.0$, the shock wave arrives at the location $r = 1.0$ with post shock density $\rho = 6.0$ and pressure $P = 0.2083$.



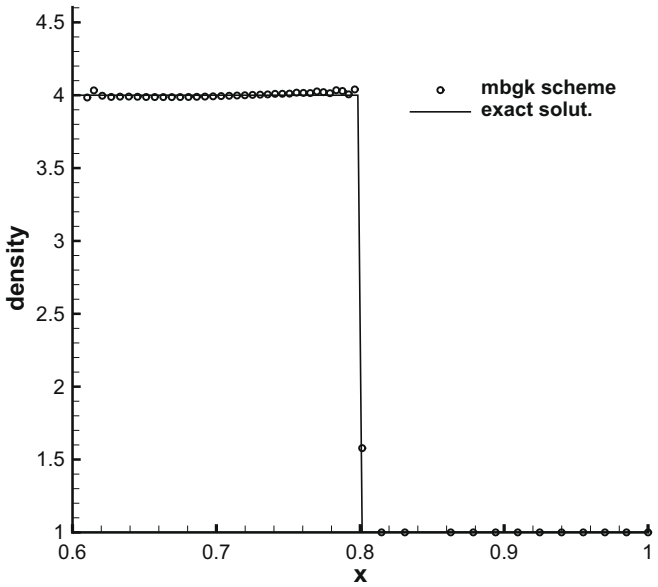


Fig. 11. Saltzman problem. Density distribution along a line of $y = 0.05$ at time $t = 0.6$. Circles are the MMBGK solution and the solid line is the exact solution.

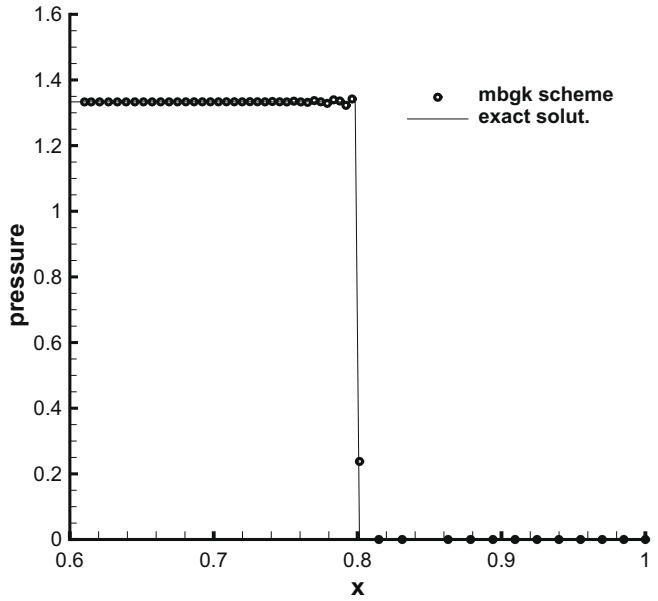


Fig. 12. Saltzman problem. Pressure distribution along a line of $y = 0.05$ at time $t = 0.6$. Circles are the MMBGK solution and the solid line is the exact solution.

We use the same idea as that presented in the last example to determine the mesh velocity, where $\mathbf{U}_g$ is a combination of the fluid velocity $\mathbf{U}$ and a diffusive velocity $\mathbf{U}_d$, i.e., $\mathbf{U}_g = \mathbf{U} + \mathbf{U}_d$. Because the flow velocity is on the radial direction only, a diffusive velocity $\mathbf{U}_d$ is also defined radially. With the density differences around the grid point (i,j),

$$\begin{cases} DL = \rho(i,j) - \rho(i-1,j-1), \\ DR = \rho(i+1,j+1) - \rho(i,j), \end{cases}$$

we define a diffusive term

$$\beta = \max\{DR - DL, 0.0\}/(DR + DL).$$

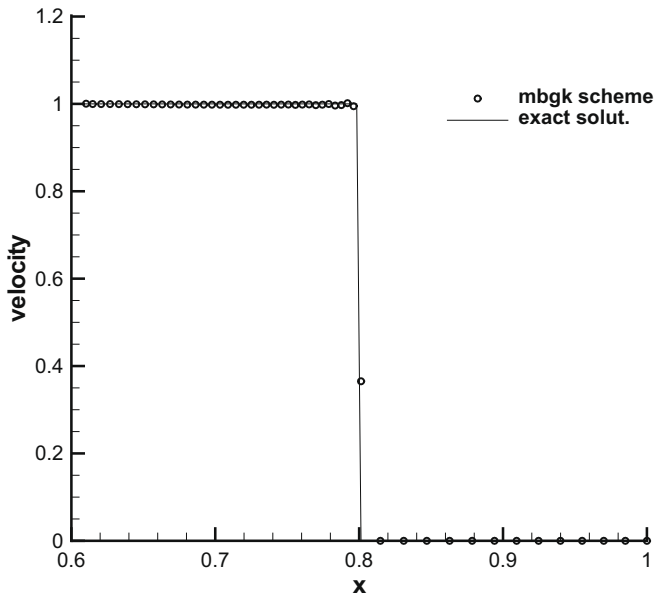


Fig. 13. Saltzman problem. Velocity distribution along a line of $y = 0.05$ at time $t = 0.6$. Circles are the MMBGK solution and the solid line is the exact solution.

Then, the diffusive mesh velocity is constructed to be $\mathbf{U}_d = \beta(1,1)^T$, and the mesh velocity becomes

$$\mathbf{U}_g = \mathbf{U} + \mu\mathbf{U}_d,$$

where $\mathbf{U}$ is the fluid velocity, and μ is a diffusion coefficient with a fixed value $\mu = 0.1$. The diffusive term is used to preserve good mesh quality as the mesh mainly follows the fluid velocity. A purely Lagrangian method for this case always results in the break-down of the computation.

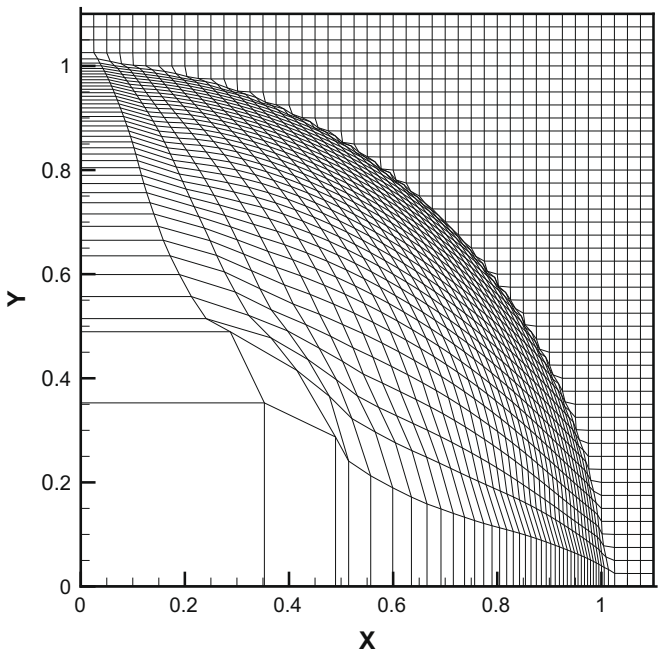


Fig. 14. Sedov problem. Mesh distributions of the MMBGK method at time $t = 1.0$.

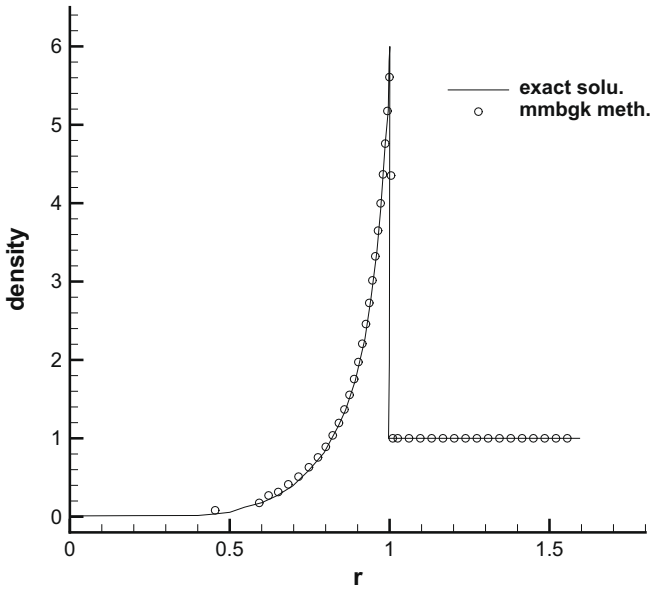


Fig. 15. Sedov problem. Density distributions along the diagonal line at time $t = 1.0$, where the distance r is given by $r = \sqrt{x(i,i)^2 + y(i,i)^2}, i = 1, \ldots, 44$. Circles are the MMBGK results with a peak value $\rho = 5.7$ and the solid line is the exact solution.

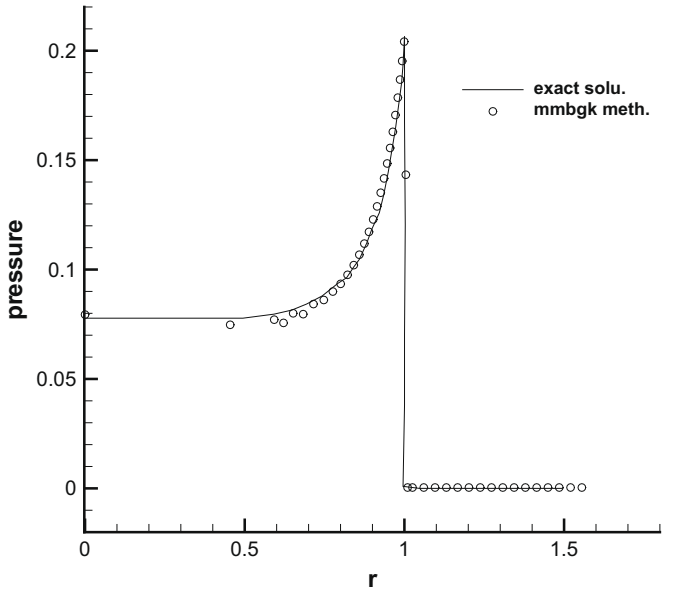


Fig. 16. Sedov problem. Pressure distributions along the diagonal line at time $t = 1.0$, where the distance r is given by $r = \sqrt{x(i,i)^2 + y(i,i)^2}, i = 1, \ldots, 44$. Circles are the MMBGK results with a peak value $p = 0.2035$ and the solid line is the exact solution.

The computational mesh at time $t = 1.0$ is shown in Fig. 14. The density profile at $t = 1.0$ is shown in Fig. 15 along a diagonal line with an angle $\theta = 45^0$ relative to x-axis, where the horizon axis in this figure is the radial distance $r = \sqrt{x(i,i)^2 + y(i,i)^2}, i = 1, \ldots, 44 (i = j)$ from the origin. The pressure profile at $t = 1.0$ is presented in Fig. 16 along the same diagonal line. At $r = 1$ the value of the exact solution for the density is 6. Fig. 15 clearly shows the good performance of the MMBGK method. The numerical results are more accurate that those presented in [24,6]. In order to check the symmetry preservation property of the scheme, the density profiles at $t = 1.0$ for all grid points are shown in Fig. 17, where the x-axis is the distance between the mesh point and the origin, which is given by $r = \sqrt{x(i,j)^2 + y(i,j)^2}, i, j = 1, \ldots, 44$. Obviously, Fig. 17 shows that a symmetry has been preserved in by our current method, which is more accurate than those presented in [7,24,6].

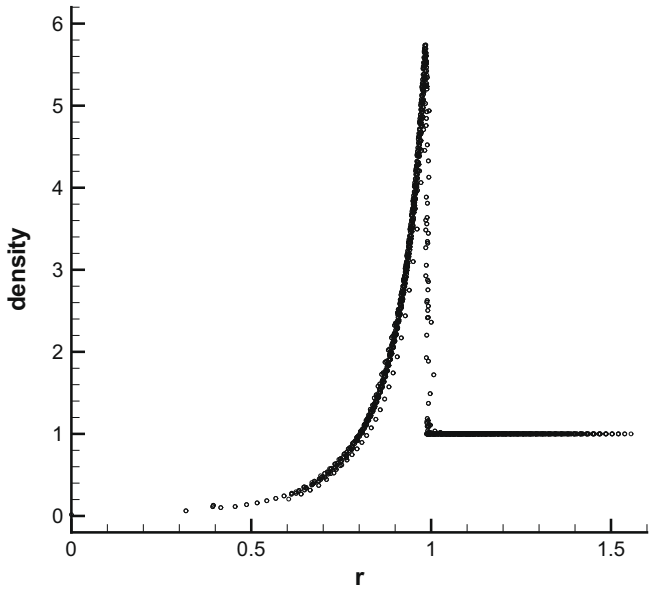


Fig. 17. Sedov problem. Density distributions for all cells at time $t = 1.0$, where the distance is defined by $r = \sqrt{x(i,j)^2 + y(i,j)^2}, i,j = 1,\ldots,44$. Perfect symmetry has been obtained by the MMBGK scheme.

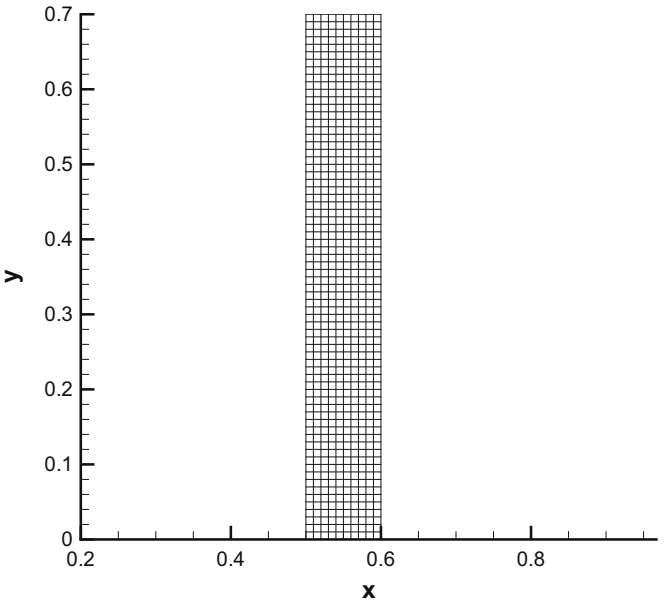


Fig. 18. Oscillating wall. Initial mesh distribution 10×70 on a domain $[0.1 \times 0.7]$.

3.3. Mesh velocity U_g determined by the moving boundary

Here, we are going to test the Navier–Stokes solution above a moving wall. Due to the specific problem, the mesh velocity follows the boundary movement, i.e., $\mathbf{U}_g = (U_g, V_g) = (U_{wall}, 0)$. Therefor, there is no mesh tangling problem.

Example 7 (*Viscous solution above an oscillating wall*). This is the well-known Stokes' second problem, which describes fluid motion above an infinite flat plate which executes sinusoidal oscillations parallel to itself. This problem has been simulated earlier by the unified moving grid gas-kinetic method [17]. The fluid above the plate is initially stationary. The governing equation of velocity U in the x-direction is

$$\frac{\partial U}{\partial t} = \nu \frac{\partial^2 U}{\partial y^2}, \tag{3.4}$$

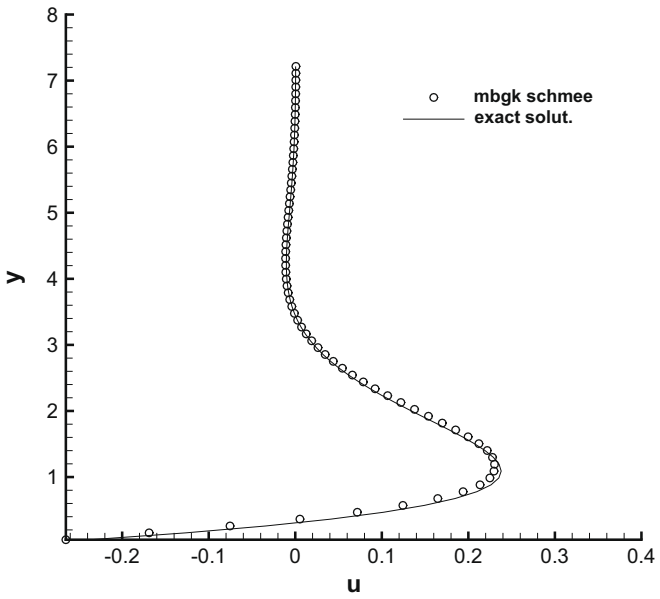


Fig. 19. Oscillating wall. Velocity distributions at time $\omega t = \pi/2$, where the circles are the results of the MMBGK method and the solid line is the exact solution.

with 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,

$$U(y,t) = U_0 e^{-y\sqrt{\omega/2\nu}} \cos\left(\omega t - y\sqrt{\frac{\omega}{2\nu}}\right). \tag{3.5}$$

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 points is used [17].

Figs. 18 and 19 show the mesh and simulation results at $\omega t = \pi/2$, where the exact solution is well captured. For a purely Lagrangian method, due to large velocity shear in the boundary layer, the Lagrangian method will stretch the mesh severely.

4. Conclusion

In this paper, a MMBGK scheme has been developed based on the integral equations of fluid dynamics. This scheme theoretically unifies the ALE method by properly choosing a mesh velocity, which not only consider the fluid velocity, but also the regularity of the mesh distribution. In other words, this paper presents a general framework, which can be used to design moving mesh method without remapping or data reconstruction between old and new meshes. The philosophy in the current research is that we need to avoid mesh distortion from the first place in the computation instead of remedying the distorted mesh afterwards. Once a mesh is distorted, the form of distortion can be infinitely diversified and it will be hard to design a universal remedying method. Since most distorted mesh cannot be precisely described by differential equations, the direct introduction of a diffusive mesh velocity on the discretized level is important, even though there may not have a unique way to design such a diffusive velocity. The current proposed methodology is useful in the unification of many existing moving mesh methods. Numerical examples in this paper demonstrate the accuracy and efficiency of the MMBGK scheme.

Acknowledgments

This work was supported by the Special Funds for Major State Basic Research Projects (Grant No. 2005CB321700), LCP (Grant No. 9140C6901010802), CAEP (Grant No. 20060604) and NSFC (Grant No. 40890154). K.X. was supported by Hong Kong Research Grant Council 621005.

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] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (1992) 235–394.
[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] Y.B. Chen, S. Jiang, An optimization-based rezoning for ALE methods, Commun. Comput. Phys. 4 (2008) 1216–1244.
[7] J. Cheng, C.W. Shu, A high order ENO conservative Lagrangian type scheme for the compressible Euler equations, J. Comput. Phys. 227 (2007) 1567–1596.
[8] J.K. Dukowicz, B.J.A. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comput. Phys. 99 (1992) 115–134.
[9] J. Glimm, X. Li, Y. Liu, Z. Xu, N. Zhao, Conservative front tracking with improved accuracy, SIAM.J. Numer. Anal. 41 (2003) 1926–1947.
[10] C.W. Hirt, A.A. Armsden, J.L. Cook, An arbitrary Lagrangian Eulerian computing method for all flow speed, J. Comput. Phys. 135 (1997) 203–216.
[11] K.A. Hoffmann, S.T. Chiang, Computational Fluid Dynamics, third ed., vol. 2, Engineering Education System, Wichita, Kansas, 1993, pp. 21–46.
[12] W. Huang, Mathematical principles of anisotropic mesh adaptation, Commun. Comput. Phys. 1 (2006) 276–310.
[13] W. H Hui, The unified coordinated system in computational fluid dynamics, Commun. Comput. Phys. 2 (2007) 577–610.
[14] 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.
[15] W.H. Hui, G.P. Zhao, Capturing contact discontinuties using the unified coordinates, in: Proceedings of Second MIT Conference on Computational Fluid and Solid Mechanics, 2003, pp. 2000–2003.
[16] C.Q. Jin, K. Xu, An adaptive grid method of two dimensional viscous flows, J. Comput. Phys. 218 (2006) 68–81.
[17] C.Q. Jin, K. Xu, An unified moving grid gas kinetic method in Eulerian space for viscous flow computation, J. Comput. Phys. 222 (2007) 155–175.
[18] C.Q. Jin, K. Xu, Numerical study of the unsteady aerodynamics of freely following plates, Commun. Comput. Phys. 3 (2008) 834–851.
[19] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
[20] L.D. Landau, E.M. Lifshitz, Fluid Mechanics, second ed., Butterworth Heinemann, 1987.
[21] P.D. Lax, X.D. Liu, Solutions of 2 dimensional Riemann problem of gas dynamical by positive schemes, SIAM J. Numer. Anal. 19 (1998) 319–340.
[22] K. Lipnikov, M. Shashkov, The error minimization based strategy for moving mesh methods, Commun. Comput. Phys. 1 (2006) 53–80.
[23] I. Lomtev, R.M. Kirby, G.E. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains, J. Comput. Phys. 155 (1999) 128–159.
[24] P.H. Maire, J. Breil, S. Galera, A cell-centred Arbitrary Lagrangian–Eulerian (ALE) method, Int. J. Numer. Meth. Fluids 56 (2008) 1161–1166.
[25] Von Neumann, R.D. Richtmyer, A method for the numerical calculations of hydrodynamics shocks, J. Appl. Phys. 21 (1950) 232–254.
[26] G.X. Ni, S. Jiang, K. Xu, Efficient kinetic scheme for steady and unsteady flow simulation on unstructured grid, J. Comput. Phys. 227 (2008) 3015–30131.
[27] G.X. Ni, S. Jiang, K. Xu, DGBGK scheme based on WENO limiters for viscous and nonviscous flow, J. Comput. Phys. 227 (2008) 799–5815.
[28] J. Qiu, C.W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiter, SIAM.J. Sci. Comput. 26 (3) (2005) 907–933.
[29] L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.
[30] W. Shyy, H.S. Udaykumar, M.M. Rao, R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, Taylor and Francis, Washington DC, 1996.
[31] 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.
[32] 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.
[33] B. Swartz, B. Wendroff, AZTEC: a front tracking code based on Godunov's method, Appl. Numer. Math. 2 (1986) 385–397.
[34] H.Z. Tang, T. Tang, Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[35] H.Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys. 1 (2006) 656–676.
[36] B. van Leer, Toward the ultimate conservative difference scheme IV, a new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[37] M.L. Wilkins, use of artificial viscosity in multidimensional fluid dynamic calculations, J. Comput. Phys. 36 (1980) 281–302.
[38] A. Winslow, Numerical solution of the quasi-linear Poisson equation, J. Comput. Phys. 1 (1967) 149–172.
[39] P. Woodward, P. Colella, The numerical simulation of two dimensional fluids with strong shock, J. Comput. Phys. 54 (1984) 115–173.
[40] K. Xu, Gas-Kinetic Schemes for Unsteady Compressible Flow Simulations, Von Kárman Institute for Fluid Dynamics, Lecture Note Series 1998-03, 1998.
[41] 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.