Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 218 (2006) 68–81
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An adaptive grid method for two-dimensional viscous flows

Changqiu Jin, Kun Xu *

Mathematics Department, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong 10000, China

Received 18 September 2005; received in revised form 23 January 2006; accepted 27 January 2006
Available online 15 March 2006

Abstract

This paper extends the gas-kinetic BGK-NS scheme to an adaptive grid for the viscous flow simulations. The grid movement and adaptation is controlled by a monitor function which may depend on velocity gradient or other flow variables, such as density or pressure. For the viscous flow computation, the use of adaptive mesh much improves the efficiency and accuracy of the method in comparison with the methods with static mesh points. The current method is an accurate and efficient method for the viscous flow computation, where the grid points can be easily moved and concentrated on the regions with large velocity and density gradients, such as the boundary layer and multi-material interface. Many numerical examples validate the current approach for the viscous flow simulations.

MSC: 65M06; 76P05; 76T05

Keywords: Adaptive grid; Gas-kinetic scheme; Navier–Stokes equations; Viscous flow

1. Introduction

In the past two decades, various adaptive grid methods have been developed for the numerical solutions in the physical and engineering applications, such as fluid dynamics, hydraulics, combustion, heat transfer, and material science. Physical problems in these areas, such as shock waves, boundary layers, and detonative waves, may require extremely fine mesh to resolve accurately large solution variations. The use of well-refined uniform mesh becomes computationally prohibitive when dealing with systems in multi-dimensions. Adaptive grid methods not only produce a high grid-point density in large gradient regions to improve the accuracy of numerical solution, but also decrease the cost of numerical calculation in comparison with the uniform grid. Currently, existing adaptive grid methods can be summarized as follows: the variational approach of Winslow [24], Brackbill et al. [3,4], Ren and Wang [17], and Tang and Tang [20]; finite element methods of Millers [15], and Davis and Flaherty [8]; moving mesh PDEs of Russell et al. [5,18], Li and Petzold [13], and Ceniceros and Hou [6]; and moving mesh methods based on harmonic mapping of Dvinsky [10] and Li et al. [12]; and many others.

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

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2006.01.041

Many existing adaptive methods are mainly targeting on the inviscid flow problems. For the viscous solution, a few finite element adaptive methods have been developed for the incompressible Navier–Stokes equations [9,16]. For the compressible flow applications, the accurate capturing of the physical structures, such as the viscous and heat conducting boundary layer and the separation bubble around an airfoil, are critically important. In these regions, large velocity and temperature gradients may exist. In order to capture these phenomena in an efficient way, an adaptive moving mesh method is an optimum choice. Therefore, it is very important to study and develop adaptive grid method for compressible viscous flow simulations.

It is well known that in the gas-kinetic theory the Navier–Stokes equations can be derived from the Boltzmann equation using the Chapman–Enskog expansion [7]. 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, the gas-kinetic BGK scheme has been well developed for the compressible viscous flow computation [26], which is specifically accurate for the supersonic viscous and heat conducting flow [27,19]. Since a gas distribution function describes both equilibrium and non-equilibrium flow properties, the inviscid and viscous fluxes are obtained simultaneously in the gas-kinetic scheme under the initial condition of a generalized Riemann solver.

This paper focuses on the development of a gas-kinetic scheme on an adaptive grid for viscous flow computation. The scheme mainly composes two parts. The first part is about the mesh redistribution and adaptation and the second part is the flow update on a newly constructed mesh. In the current paper, the former one is based on the adaptive method proposed by Tang and Tang [20]. Since the velocity is an important quantity for the viscous flow, we are going to use velocity gradient to define the monitor function, then the mesh will be redistributed according to this function by solving elliptic equations. In the second part, the gas-kinetic BGK-NS scheme for the Navier–Stokes fluxes will be used to update the flow variables for each control volume [26], where a time-dependent gas distribution function is obtained from an initially discontinuity flow distribution.

This paper is organized in the following. In the next section, we are going to outline the theory of mesh-redistribution method. Section 3 presents the gas-kinetic BGK scheme and its use in the adaptive framework. Section 4 presents many numerical examples to validate the accuracy and efficiency of the newly developed scheme. The last section is the conclusion.

2. A mesh-redistribution method

This section contains two subsections, which are mesh generation based on the variational approach and conservative flow quantities redistribution to the newly generated mesh.

2.1. Mesh generation based on the variational approach

In one-dimensional space (1D), due to the capability of using fine mesh in the current computing power, moving mesh method is generally not necessary. Here, in this paper, we will focus on the two-dimensional space (2D). In 2D, the widely used mesh generation techniques are based on the variational approaches. The pioneer work was proposed by Winslow [24]. Let $\mathbf{x} = (x, y)$ 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

$$x = x(\xi, \eta), \quad y = y(\xi, \eta), \tag{1}$$

and the inverse map is

$$\xi = \xi(x, y), \quad \eta = \eta(x, y). \tag{2}$$

The specific map is obtained by minimizing of a mesh adaptation functional of the following form

$$E(\xi, \eta) = \frac{1}{2} \int_{\Omega_p} (\nabla \xi^T G_1^{-1} \nabla \xi + \nabla \eta^T G_2^{-1} \nabla \eta) \, dx \, dy, \tag{3}$$

where G_1 and G_2 are symmetric positive definite matrices which are formally called monitor functions. The Euler–Lagrange equations for $E(\xi, \eta)$ become,

$$\nabla \cdot (G_1^{-1}\nabla\xi) = 0, \quad \nabla \cdot (G_2^{-1}\nabla\eta) = 0, \tag{4}$$

which gives the coordinate transformation in mesh generation and adaptation. Grid generation is basically to obtain the curvilinear coordinate system (1) from the above elliptic system (4). Usually, after solving the linear system (4) for $\boldsymbol{\xi}(\boldsymbol{x})$, we find the inverse map to obtain $\boldsymbol{x}(\boldsymbol{\xi})$, which is expensive. Certainly, we can directly solve the corresponding equations on the computational domain Ω_c by interchanging the dependent and independent variables in (4). However, the obtained equations are complicated and massive computations are required. An alternative approach, as suggested by Ceniceros and Hou [6], is to consider a functional defined in the computational domain:

$$\tilde{E}[\xi,\eta] = \frac{1}{2}\int_{\Omega_c} (\tilde{\nabla}^{\mathrm{T}} x G_1 \tilde{\nabla} x + \tilde{\nabla}^{\mathrm{T}} y G_2 \tilde{\nabla} y)\,\mathrm{d}\xi\,\mathrm{d}\eta, \tag{5}$$

to replace the conventional functional (3), where $\tilde{\nabla} = (\partial_\xi, \partial_\eta)^{\mathrm{T}}$. The corresponding Euler–Lagrange equations become

$$\tilde{\nabla} \cdot (G_1\tilde{\nabla} x) = 0, \quad \tilde{\nabla} \cdot (G_2\tilde{\nabla} y) = 0. \tag{6}$$

Therefore, the mesh distribution in the physical space can be directly obtained by solving (6), which is much simpler than the conventional variational approach (3). However, the system (6) can generate degenerating grids in some concave regions [10]. The original system (4) is more accurate and reliable than the simple one (6) even though it is more complicated. In the current paper, all numerical examples have simple geometry, so Eq. (6) will be used for the mesh generation.

In the present approach, choosing an appropriate monitor function is very important in the mesh generation and adaptation. In this paper, we will use a directional splitting monitor function, i.e., $G_1 = G_2 = \mathrm{diag}\{w_1, w_2\}$, where the functions w_1, w_2 are defined by

$$w_1 = \left(1 + \alpha_1|\psi|^2 + \beta_1\left|\frac{\partial\psi}{\partial\xi}\right|^2\right)^{\frac{\gamma_1}{2}}, \quad w_2 = \left(1 + \alpha_2|\psi|^2 + \beta_2\left|\frac{\partial\psi}{\partial\eta}\right|^2\right)^{\frac{\gamma_2}{2}}, \tag{7}$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2$ are some nonnegative constants, and their optimum values depend on flow problems. In this paper, the selection of these parameters is mainly based on the physical intuition and numerical testing. Here ψ can be chosen as density, velocity, entropy, or error estimate. But, for the viscous flow, the use of flow velocity seems to be a good choice.

The specific steps for constructing an adaptive mesh are the following. We firstly divide the computational domain $\Omega_c = \{(\xi,\eta)|1 \geqslant \xi,\eta \geqslant 0\}$ into the square mesh:

$$\{(\xi_j,\eta_k)|\xi_j = \frac{j}{J_x+1}, \quad \eta_k = \frac{k}{J_y+1}; \quad 0 \leqslant j \leqslant J_x+1,\ 0 \leqslant k \leqslant J_y+1\},$$

then, a second-order central difference scheme is used to discretize the mesh generation Eq. (6)

$$\begin{aligned}
&\frac{\Delta_-^j((w_1)_{j+\frac{1}{2},k}\Delta_+^j x_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k((w_2)_{j,k+\frac{1}{2}}\Delta_+^k x_{j,k})}{(\Delta\eta)^2} = 0,\\
&\frac{\Delta_-^j((w_1)_{j+\frac{1}{2},k}\Delta_+^j y_{j,k})}{(\Delta\xi)^2} + \frac{\Delta_-^k((w_2)_{j,k+\frac{1}{2}}\Delta_+^k y_{j,k})}{(\Delta\eta)^2} = 0,
\end{aligned} \tag{8}$$

where Δ_+^l and Δ_-^l, $l = j, k$ denote forward and backward difference operators. The mesh mapping (1) are denoted by $x_{j,k} = x(\xi_{j,k},\eta_{j,k})$, $y_{j,k} = y(\xi_{j,k},\eta_{j,k})$. The above equations with boundary condition are solved by iterative methods, i.e., Jacobi iteration or Gauss–Seidel method. The numerical experiments in Section 4 show that the Gauss–Seidel iteration with a fixed number of iteration is very robust in the mesh construction.

2.2. Conservative solution interpolation

After generating new mesh at each time step according to the monitor function, we need to interpolate flow variables from the old to the newly obtained mesh. Even though many interpolating schemes have been sug-

gested, such as the non-conservative one for the nonlinear Hamilton–Jacobi equation [21], for the compressible flow equations a conservative interpolation scheme has to be used. In the current research, the method proposed by Tang and Tang [20] for a finite volume formulation is adopted. Let $(x_{j,k}, y_{j,k})$ and $(\tilde{x}_{j,k}, \tilde{y}_{j,k})$ be coordinates of the old and new grid points, respectively, and at the same time $\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}$ and $A_{j+\frac{1}{2},k+\frac{1}{2}}$ denote the quadrangles with four vertices $(\tilde{x}_{j+p,k+q}, \tilde{y}_{j+p,k+q})$, and $(x_{j+p,k+q}, y_{j+p,k+q})$, p, q are integer numbers with $0 \leqslant p, q \leqslant 1$.

A perturbation method is applied in the derivation of the conservative interpolation scheme. Assuming $(\tilde{x}, \tilde{y}) = (x - c^x(x,y), y - c^y(x,y))$ with small displacement (c^x, c^y), we have

$$\begin{aligned}
\int_{\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}} U(\tilde{x},\tilde{y})\,\mathrm{d}\tilde{x}\,\mathrm{d}\tilde{y} &= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x-c^x, y-c^y)\det\left(\frac{\partial(\tilde{x},\tilde{y})}{\partial(x,y)}\right)\mathrm{d}x\,\mathrm{d}y \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} (U(x,y) - c^x U_x - c^y U_y)(1 - c^x_x - c^y_y)\,\mathrm{d}x\,\mathrm{d}y \\
&\approx \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} [U(x,y) - (c^x U)_x - (c^y U)_y]\,\mathrm{d}x\,\mathrm{d}y \\
&= \int_{A_{j+\frac{1}{2},k+\frac{1}{2}}} U(x,y)\,\mathrm{d}x\,\mathrm{d}y - [(c_n^2 U)_{j+1,k+\frac{1}{2}} + (c_n^4 U)_{j,k+\frac{1}{2}}] - [(c_n^3 U)_{j+\frac{1}{2},k+1} + (c_n^1 U)_{j+\frac{1}{2},k}], \qquad (9)
\end{aligned}$$

where both the coordinate transformation from $(\tilde{x}, \tilde{y})$ to (x, y) and the Taylor expansion of $U(\tilde{x}, \tilde{y})$ at (x, y) are used in the above derivation. Higher order terms have been neglected. Here $c_n^l := c^x n_x^l + c^y n_y^l$ for $l = 1, 2, 3, 4$, where $n^l = (n_x^l, n_y^l)$ is the unit outward normal direction on the corresponding surface of the control volume $A_{j,k}$. More detailed explanation can be found in [20]. From (9), the conservative quantities are interpolated into the new mesh with the value,

$$|\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} = |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}} - [(c_n^2 U)_{j+1,k+\frac{1}{2}} + (c_n^4 U)_{j,k+\frac{1}{2}}] - [(c_n^3 U)_{j+\frac{1}{2},k+1} + (c_n^1 U)_{j+\frac{1}{2},k}]. \qquad (10)$$

Taking the summation on both sides in the above equation for j, k, the conservative property of the above scheme can be proved, i.e.,

$$\sum_{j,k} |\tilde{A}_{j+\frac{1}{2},k+\frac{1}{2}}|\tilde{U}_{j+\frac{1}{2},k+\frac{1}{2}} = \sum_{j,k} |A_{j+\frac{1}{2},k+\frac{1}{2}}|U_{j+\frac{1}{2},k+\frac{1}{2}}. \qquad (11)$$

3. Gas kinetic BGK scheme

After interpolating the flow variables from the old to the new mesh, as any other finite volume method the fluxes will be evaluated at a cell interface to update the solution to the next time level. In the current paper, a gas-kinetic scheme will be used to construct the numerical flux at the cell interface, where the inviscid and viscous parts are obtained simultaneously.

3.1. Initial reconstruction

Firstly, we need to reconstruct the initial data at each cell from the cell averages. A nonlinear limiter, such as van Leer limiter [22], can be used here for the subcell data reconstruction at the beginning of each time step. Let denote $X_{j,k} = (x_{j,k}, y_{j,k})$ as the center coordinate of the cell (j, k). For a directional splitting scheme, such as in the j-direction, we have both cell-averaged value $w_{j,k}^n$ and those $w_{j-1,k}^n, w_{j+1,k}^n$ in its neighboring cells. The limited slope becomes

$$L(s_+, s_-) = (\mathrm{sign}(s_+), \mathrm{sign}(s_-))\frac{|s_-||s_+|}{|s_+| + |s_-|},$$

where $s_+ = (w_{j+1,k}^n - w_{j,k}^n)/(x_{j+1,k} - x_{j,k})$ and $s_- = (w_{j,k}^n - w_{j-1,k}^n)/(x_{j,k} - x_{j-1,k})$. Then, the reconstructed initial flow variable W^n inside cell (j, k) has the form

$$\bar{W}^n_{j,k}(x,y) = W^n_{j,k} + L(s_+, s_-)\cdot(X - X_{j,k}), \quad (x,y) \subseteq A_{j,k}, \tag{12}$$

where $X = (x, y)$. In the following, the gas-kinetic BGK Navier–Stokes (BGK-NS) solver will be presented for the NS flux evaluation at the cell interface $j + 1/2$.

3.2. Gas-kinetic BGK-NS flow solver

The BGK model of the approximate Boltzmann equation can be written as [1]

$$f_t + uf_x + vf_y = \frac{g - f}{\tau}, \tag{13}$$

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 velocities (u, v), and internal variable ξ. The particle collision time τ is related to the viscosity and heat conduction coefficients. The equilibrium state is a Maxwellian distribution,

$$g = \rho\left(\frac{\lambda}{\pi}\right)^{\frac{K+2}{2}} \mathrm{e}^{-\lambda((u-U)^2+(v-V)^2+\xi^2)},$$

where ρ is the density, U and V are the macroscopic velocities in the x and y directions, and λ is related to the gas temperature $m/2k\ T$. For a 2D flow, the particle motion in the z direction is included into the internal variable ξ, and the total number of degrees of freedom K in ξ is equal to $(5 - 3\gamma)/(\gamma - 1) + 1$. In the equilibrium state, ξ^2 is equal to $\xi^2 = \xi_1^2 + \xi_2^2 + \cdots + \xi_K^2$. The relation between mass ρ, momentum $(n = \rho U, m = \rho V)$, and energy E densities with the distribution function f is

$$w = \begin{pmatrix} \rho \\ n \\ m \\ E \end{pmatrix} = \int \psi_\alpha f \,\mathrm{d}\Xi, \quad \alpha = 1, 2, 3, 4, \tag{14}$$

where ψ_α is the component of the vector of moments

$$\boldsymbol{\psi} = (\psi_1, \psi_2, \psi_3, \psi_4)^{\mathrm{T}} = (1, u, v, \frac{1}{2}(u^2 + v^2 + \xi^2))^{\mathrm{T}},$$

and $\mathrm{d}\Xi = \mathrm{d}u\,\mathrm{d}v\,\mathrm{d}\xi$ is the volume element in the phase space with $\mathrm{d}\xi = \mathrm{d}\xi_1, \mathrm{d}\xi_2, \ldots, \mathrm{d}\xi_K$. Since mass, momentum and energy are conserved during particle collisions, f and g satisfy the conservation constraint

$$\int (g - f)\psi_\alpha \,\mathrm{d}\Xi = 0, \quad \alpha = 1, 2, 3, 4, \tag{15}$$

at any point in space and time. The derivation from the BGK model to the NS equations for a polyatomic gases can be found in [25].

The general solution f of the BGK model at a cell interface $(x_{j+1/2,k}, y_{j+1/2,k})$ and time t is

$$f(x_{j+1/2,k}, y_{j+1/2,k}, t, u, v, \xi) = \frac{1}{\tau}\int_0^t g(x', y', t', u, v, \xi)\mathrm{e}^{-(t-t')/\tau}\,\mathrm{d}t' + \mathrm{e}^{-t/\tau} f_0(x_{j+1/2,k} - ut, y_{j+1/2,k} - vt), \tag{16}$$

where $x' = x_{j+1/2,k} - u(t - t')$, $y' = y_{j+1/2,k} - v(t - t')$ are the trajectory of a particle motion and f_0 is the initial gas distribution function f at the beginning of each time step $(t = 0)$. Two unknowns g and f_0 must be specified in Eq. (16). In order to simplify the notation, $(x_{j+1/2,k} = 0, y_{j+1/2,k} = 0)$ will be used in the following text.

The directional splitting BGK-NS method is described in [26]. For a cell interface $(j + 1/2, k)$, the normal direction is assumed to be in the x-direction. The initial gas distribution function f_0 has the form,

$$f_0 = \begin{cases} g^l\big(1 + a^l x - \tau(a^l u + A^l)\big), & x \leqslant 0, \\ g^r(1 + a^r x - \tau(a^r u + A^r)), & x \geqslant 0. \end{cases} \tag{17}$$

The terms proportional to τ represent the non-equilibrium parts in the Chapman–Enskog expansion of the BGK model, which have no direct contribution to the conservative flow variables, i.e.,

$$\int (a^l u + A^l)\psi_\alpha g^l \,\mathrm{d}\Xi = 0,$$
$$\int (a^r u + A^r)\psi_\alpha g^l \,\mathrm{d}\Xi = 0. \tag{18}$$

After having f_0, the equilibrium state g around $(x = 0, t = 0)$ is constructed by

$$g = g_0\big(1 + (1 - H[x])\bar{a}^l x + H[x]\bar{a}^r x + \bar{A}t\big), \tag{19}$$

where $H[x]$ is the Heaviside function defined as

$$H[x] = \begin{cases} 0, & x < 0, \\ 1, & x \geqslant 0. \end{cases}$$

Here g_0 is a local Maxwellian distribution function located at $(x = 0, y = 0)$. In both f_0 and g, $a^l, A^l, a^r, A^r, \bar{a}^l, \bar{a}^r$, and $\bar{A}$ are related to the derivatives of a Maxwellian distribution function in space and time. The dependence of $a^l, a^r, \ldots, \bar{A}$ on the particle velocities can be obtained from a Taylor expansion of a Maxwellian [26].

After determining all parameters in the initial gas distribution function f_0 and the equilibrium state g, we can get the solution of f at time t at the cell interface

$$\begin{aligned} f(x_{j+1/2,k}, y_{j+1/2,k}, t, u, v, \xi) = {} & (1 - \mathrm{e}^{-t/\tau})g_0 + (\tau(-1 + \mathrm{e}^{-t/\tau}) + t\mathrm{e}^{-t/\tau})(\bar{a}^l \mathrm{H}[u] + \bar{a}^r(1 - \mathrm{H}[u]))ug_0 \\ & + \tau(t/\tau - 1 + \mathrm{e}^{-t/\tau})\bar{A}g_0 + \mathrm{e}^{-t/\tau}((1 - (t + \tau)ua^l)\mathrm{H}[u]g^l + (1 - (t + \tau)ua^r) \\ & \times (1 - \mathrm{H}[u])g^r) + \mathrm{e}^{-t/\tau}(-\tau A^l \mathrm{H}[u]g^l - \tau A^r(1 - \mathrm{H}[u])g^r). \end{aligned} \tag{20}$$

The time-dependent numerical fluxes in the normal-direction across the cell interface can be computed by

$$\begin{pmatrix} \mathscr{F}_\rho \\ \mathscr{F}_m \\ \mathscr{F}_n \\ \mathscr{F}_E \end{pmatrix}_{j+1/2} = \int u \begin{pmatrix} 1 \\ u \\ v \\ \frac{1}{2}(u^2 + v^2 + \xi^2) \end{pmatrix} f(x_{j+1/2,k}, y_{j+1/2,k}, t, u, v, \xi)\,\mathrm{d}\Xi, \tag{21}$$

where $f(x_{j+1/2,k}, y_{j+1/2,k}, t, u, v, \xi)$ is given by Eq. (20). By integrating the above equation to the whole time step, we can get the total mass, momentum and energy transport across a cell interface. Also, the unit Prandtl number can be changed to any value by modifying the energy flux in the above equation [26].

4. Numerical experiments

The gas-kinetic scheme is a Navier–Stokes solver, the physical viscosity coefficient is proportional to the particle collision time. The inviscid flow is considered as a limiting solution when the viscosity coefficient is small. In this section, both inviscid and viscous flow test cases will be presented by the current adaptive method.

4.1. Inviscid flow

For the inviscid flow computation, the collision time τ in the gas-kinetic flux function is defined as

$$\tau = 0.05\Delta t + \frac{|P_l - P_r|}{P_l + P_r}\Delta t,$$

where Δt is the CFL time step, and P_l, P_r are the corresponding pressure in the states g_l, g_r of the initial discontinuous gas distribution function f_0. Due to the large density gradient at the material interface, in the following example the density is used in the monitor function (7).

Case(1) A Mach = 1.22 shock wave in air hitting a helium cylindrical bubble

A Mach *1.22* planar shock wave, moving through air, impinges on a cylindrical helium bubble. The flow is often modelled by two-dimensional compressible Euler equations of a two-component fluid (air–helium). The

earlier result using the gas-kinetic BGK scheme has been reported in [14]. Here, we mainly demonstrate that the adaptive grid method can give much high resolution solution with less grid points, especially in the regions with large density gradient, such as the shock waves and the air–helium interface. The initial condition for this case is

$$(\rho, U, V, P, \gamma) = \begin{cases} (1, 0, 0, 1, 1.4), & \text{pre-shock air,} \\ (1.3764, -0.394, 0, 1.5698, 1.4), & \text{post-shock air,} \\ (0.1358, 0, 0, 1, 1.67), & \text{helium.} \end{cases}$$

In the case, inflow and outflow boundary conditions are used on the right and left sides of the computational domain, and reflection conditions are imposed on the top and bottom.

The computational mesh used in the current adaptive method is 260×71 which is only 4% of the total number of uniform grids 1300×356 used in [14]. Fig. 1 is the adaptive grids from the monitor function (7) with $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1.0$ at time $t = 150.0$. With the similar resolution at the material interface, the total computational time using the adaptive method is less than 30% of the one used in a uniform fine mesh 1300×356. The density distribution is shown in Fig. 2, which clearly shows the coupling between the computational mesh and flow resolution. As expected, the adaptive grid method improves the interface resolution greatly in comparison with the uniform mesh result in Fig. 3, where the same number of grid points are used as Fig. 2.

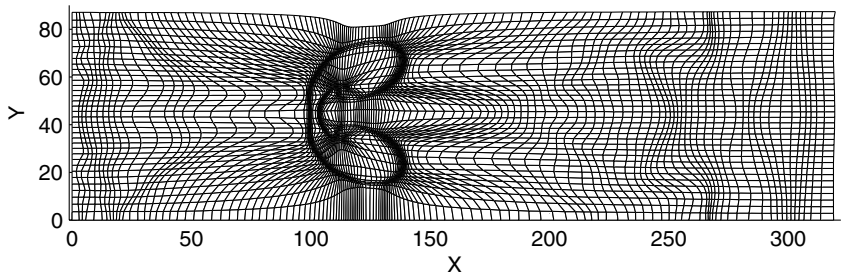


Fig. 1. Adaptive mesh distribution at time $t = 150.00$.

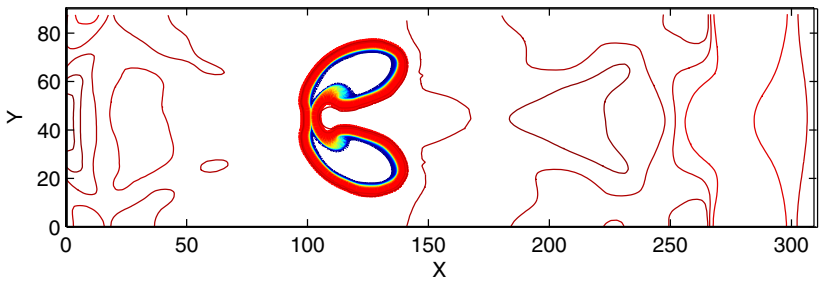


Fig. 2. Density contours on the adaptive mesh at time $t = 150.0$.

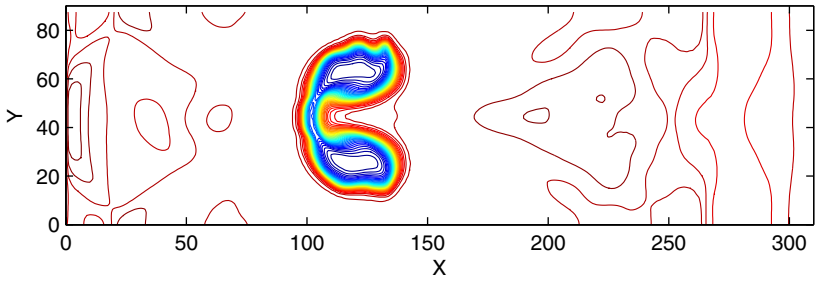


Fig. 3. Density contours on a uniform mesh at time $t = 150.0$.

4.2. Viscous flows

In the viscous flow calculations, the collision time τ is set to be $\tau = \mu/P$, or $\tau = 2v\lambda$ with $\lambda = \rho/2P$, where μ and v are the dynamical and kinematic viscosity coefficients, ρ is the fluid density, and P is the local pressure. For the viscous flow simulation, the viscous effect is closely related to the velocity gradient. Therefore, ψ in the monitor function (7) will take the fluid velocity $q = \sqrt{U^2 + V^2}$.

Case(2) Flow above an oscillating plate

This is called Stokes's second problem, which considers fluid motion above an infinite flat plate which executes sinusoidal oscillations parallel to itself. The fluid above the plate is initially stationary. The governing equation of velocity U in the x-direction is

$$\frac{\partial U}{\partial t} = v\frac{\partial^2 U}{\partial y^2}, \tag{22}$$

with the boundary conditions

$$U(0,t) = U_0 \cos \omega t, \quad U(\infty, t) = 0.$$

The exact solution for the above problem is,

$$U = U_0 e^{-y\sqrt{\omega/2v}} \cos\left(\omega t - y\sqrt{\frac{\omega}{2v}}\right). \tag{23}$$

At $y = 4\sqrt{v/\omega}$, 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 $\delta \sim 4\sqrt{v/\omega}$. 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 $v = 0.00046395$, and a mesh size 10×20 is used in the computation. Obviously, we only need to move the mesh in the y direction, where $\alpha_2 = 0, \beta_2 = 1.0$ are used in the corresponding monitor function. Fig. 4 shows the two adaptive grids at times $\omega t = \frac{\pi}{10}$ and $\frac{\pi}{2}$. The numerical results are compared with the exact solutions (23) in Fig. 5. Certainly, the grid points concentrate in the regions with large velocity gradient. The results clearly show that the adaptive mesh method is very effective to capture the viscous solution.

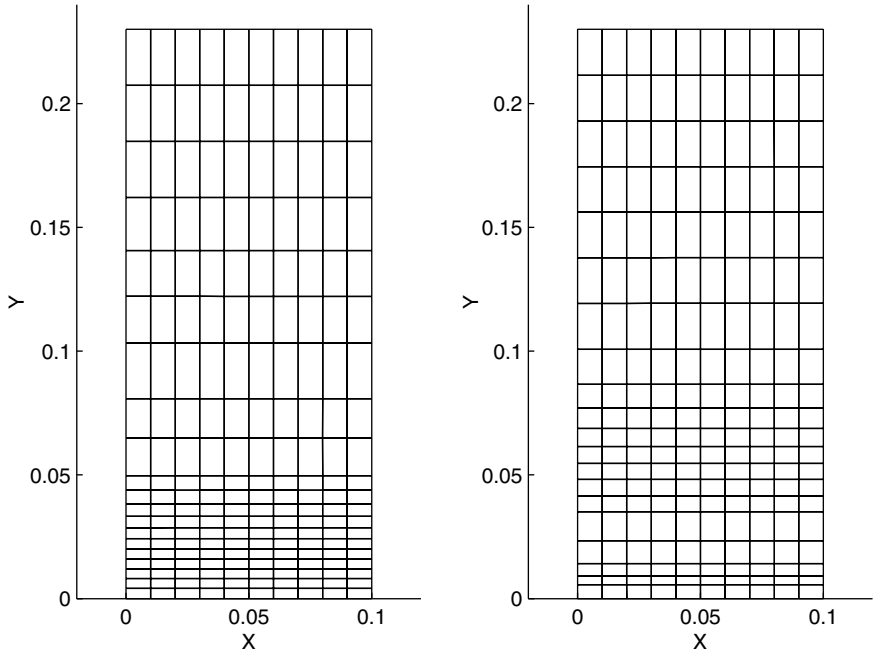


Fig. 4. Adaptive mesh at $\omega t = \frac{\pi}{10}$ (left) and $\frac{\pi}{2}$ (right).

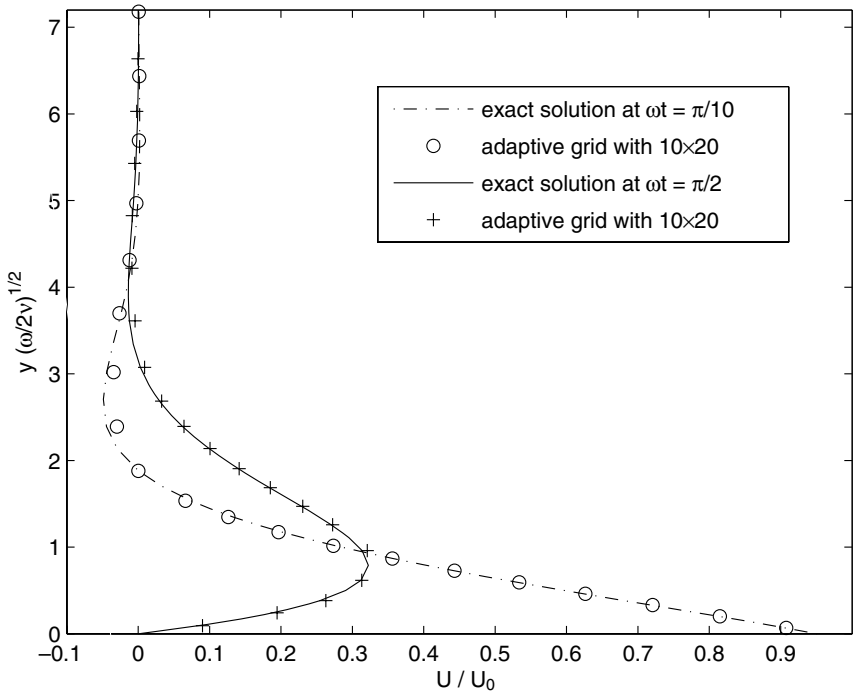


Fig. 5. Velocity distribution at $\omega t = \frac{\pi}{10}$ and $\frac{\pi}{2}$.

Case(3) Shock reflection from a wedge

This is an unsteady Mach reflection problem, where a plane shock with Mach number $M = 1.3$ moves from left to right across a wedge with 25°. The initial flow state is given by $(p, \rho, u, v) = (1/1.4, 1, 0, 0)$ inside the computational domain, and the left boundary condition with

$$(p, \rho, u, v) = (1.2893, 1.5157, 0.44231, 0)$$

satisfies the shock jump condition. Non- slip boundary condition is imposed on the surface of the wedge. Initially, the computational domain $\{0 \leqslant x \leqslant 2.0, 0 \leqslant y \leqslant 0.75\}$ is covered by a uniform mesh with 200×200 grid

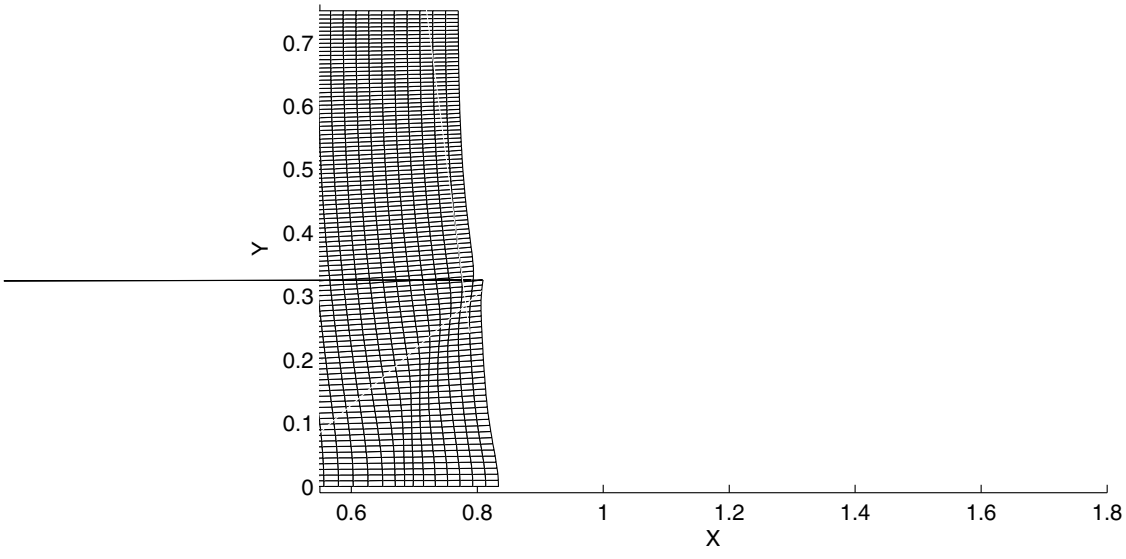
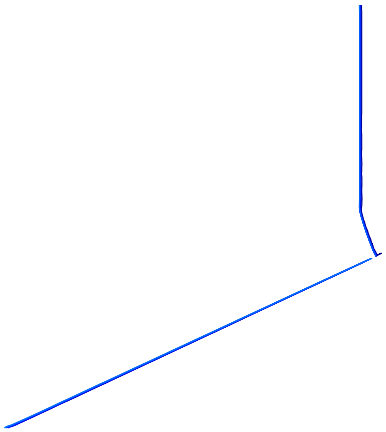


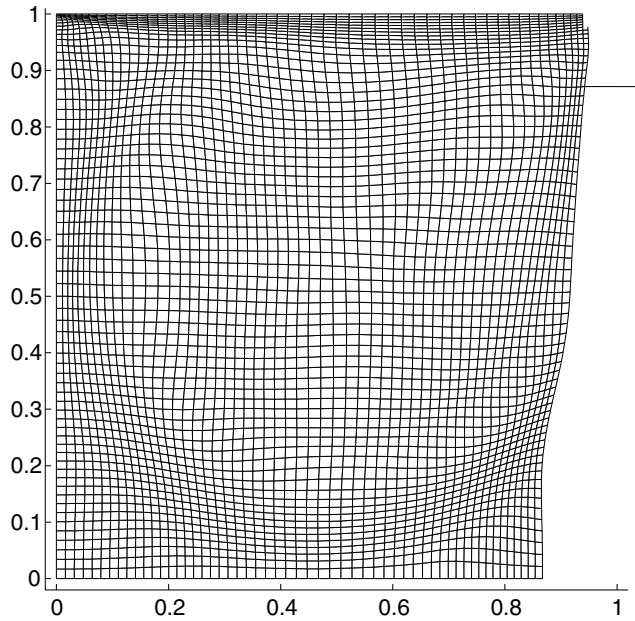
Fig. 6. Adaptive mesh after the shock reflection from a wedge.

points and cell size $\Delta x = 0.01$ and $\Delta y = 0.00375$. The kinematic viscosity coefficient takes a value $v = 0.0000369$. The adaptive mesh is constructed with the parameters $\alpha_1 = \alpha_2 = 0$ and $\beta_1 = \beta_2 = 1.0$ in the monitor function. With the flow movement, the mesh starts to concentrate in the regions with high velocity gradients, such as the shock front and boundary layer. Fig. 6 shows the adaptive mesh in the physical domain. The Mach number contours at time $t = 1.25$ are shown in Fig. 7, which is close the experiment picture in [23].

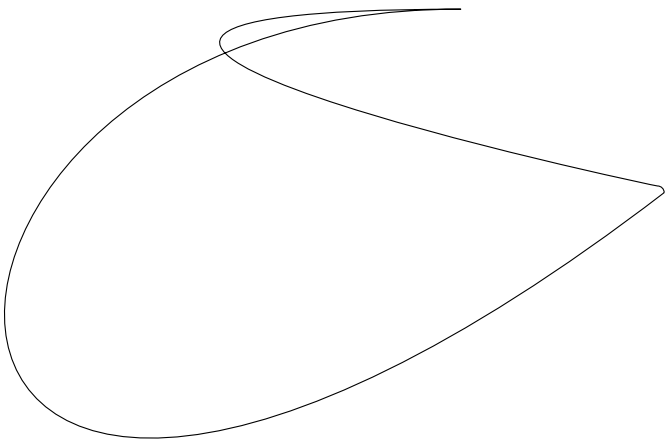
In order to demonstrate the accuracy of the mesh moving method, we also used a much refined uniform mesh with 200×1000 grid points to do the same calculation. The boundary layer profile at the location $x = 1.175$ above the wedge surface is presented in Fig. 8. Obviously, the results from the adaptive mesh



method is consistent with the refinement results, where the results with uniform coarse mesh (200×200 grid points) are inferior in comparison with others. In other words, the adaptive grid method with much less number of grid points can obtain accurate solution. The adaptive mesh method (200×200 grid points) uses less than 30% of computational time than that with a uniform refined mesh (200×1000 grid points).



XYFig. 9. Adaptive mesh for the cavity flow.



0.500.5100.10.20.30.40.50.60.7

Case(4) 2D cavity flow

The cavity flow inside a square is a well-defined example for validating numerical methods. There is highly accurate solutions for this case, which were obtained by Ghia et al. [11] and Botella and Peyret [2]. The square

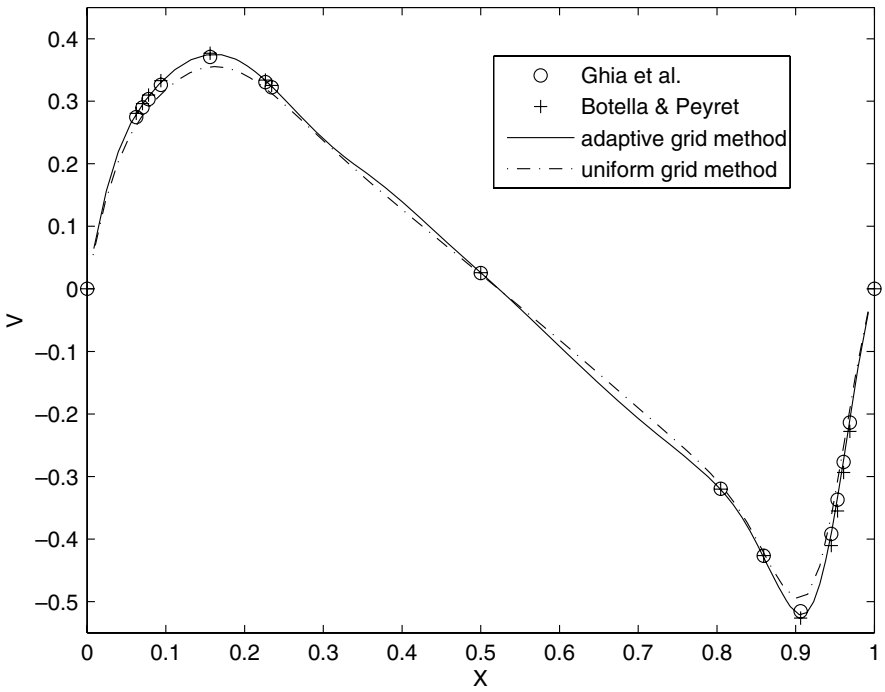


Fig. 11. Velocity V profile along the central horizontal line at $Re = 1000$.

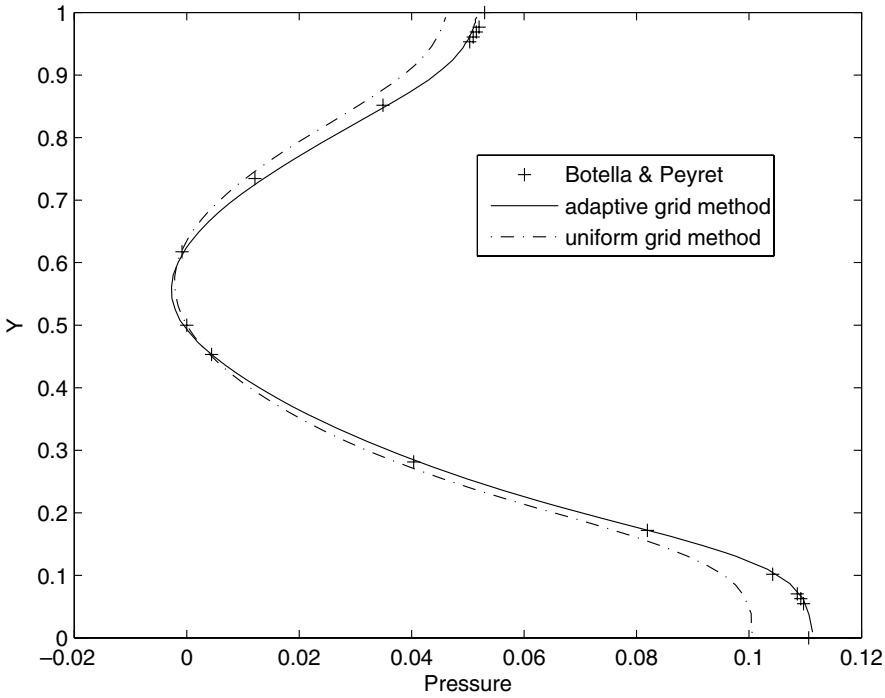


Fig. 12. Pressure profile along the central vertical line at $Re = 1000$.

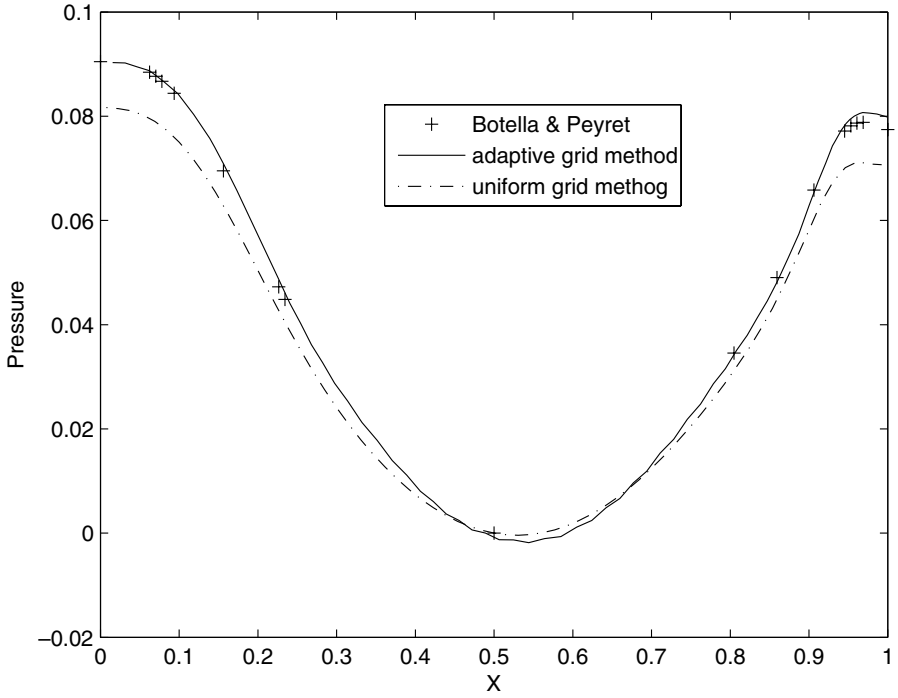


Fig. 13. Pressure profile along the central horizontal line at $Re = 1000$.

has three fixed walls and the upper moving boundary with unit velocity. Hence, the non-slip boundary condition relative to the walls are imposed. In the current simulation, we only calculate the flow at Reynolds number $\mathrm{Re} = 1000$ with 65×65 grid points in the computational domain. The mesh adaptation is done with $\alpha_1 = \alpha_2 = 0$, $\beta_1 = \beta_2 = 1.5$ in the monitor function. Fig. 9 shows the adaptive grids, where the mesh points concentrate on the high velocity gradient regions. The simulation results, as well as the benchmark solutions from Ghia et al. [11] and Botella and Peyret [2], are presented in Figs. 10–13. The solutions from the adaptive method have better agreement with the benchmark results than that from uniform mesh. With similar accuracy, the computational time here only takes 23.5% of that used in the uniform mesh method with 255×255 grid points in [28].

5. Conclusion

In this paper, an adaptive grid method has been successfully developed for the simulation of inviscid and viscous flows. For unsteady flow computation, the adaptive method has obvious advantage in terms of the accuracy and efficiency in comparison with the static mesh method. The good agreement between the numerical results and benchmark solutions validates the current approach. With the velocity-gradient dependence monitor function, the boundary layer and shock wave have been well captured for the viscous flow. In the future, with further development the adaptive mesh method will become an important working horse for the unsteady viscous flow computation, especially in the cases where fine grid resolution is needed, such as DNS and turbulent flow.

Acknowledgements

The authors thank Prof. H.Z. Tang for helpful discussion and reviewers for their constructive comments. 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 Nos. HKUST6102/04E, 6210/05E, and HIA04/05.EG02).

References

[1] 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.
[2] O. Botella, R. Peyret, Benchmark spectral results on the lid-driven cavity flow, Comput. Fluids 27 (1998) 421–433.
[3] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[4] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.
[5] 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.
[6] H.D. Ceniceros, T.Y. Hou, An efficient dynamically adaptive mesh for potentially singular solutions, J. Comput. Phys. 172 (2001) 609–639.
[7] S. Chapman, T.G. Cowling, The Mathematical Theory of Non-uniform Gases, Cambridge University Press, 1990.
[8] S.F. Davis, J.E. Flaherty, An adaptive finite element method for initial-boundary value problems for partial differential equations, SIAM J. Sci. Stat. Comput. 3 (1982) 6–27.
[9] Y. Di, R. Li, T. Tang, P. Zhang, Moving mesh finite element methods for incompressible Navier–Stokes equations, J. Comput. Phys. 190 (2003) 100–117.
[10] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 221–244.
[11] 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.
[12] R. Li, T. Tang, P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[13] S. Li, L. Petzold, Moving mesh methods with upwinding schemes for time-dependent PDEs, J. Comput. Phys. 131 (1997) 368–377.
[14] Y.S. Lian, K. Xu, A gas-kinetic scheme for multimaterial flows and its application in chemical reactions, J. Comput. Phys. 163 (2000) 349–375.
[15] K. Miller, R.N. Miller, Moving finite element, I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[16] B. Perot, R. Nallapati, A moving unstructured staggered mesh method for the simulation of incompressible free-surface flows, J. Comput. Phys. 190 (2003) 100–117.
[17] W.Q. Ren, X.P. Wang, An iterative grid redistribution method for singular problems in multiple dimensions, J. Comput. Phys. 159 (2000) 246–273.
[18] 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.
[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 (2005) 313–331.
[20] H.Z. Tang, T. Tang, Adaptive mesh methods for one-and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[21] H.Z. Tang, T. Tang, P.W. Zhang, Adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions, J. Comput. Phys. 188 (2003) 543–572.
[22] B. van Leer, Toward the ultimate conservative difference scheme IV, A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[23] M. van Dyke, An Album of Fluid Motion, The Parabolic Press, Stanford, CA, 1982, p. 143.
[24] A. Winslow, Numerical solution of the quasi-linear Poisson equation, J. Comput. Phys. 1 (1967) 149–172.
[25] K. Xu, Gas-kinetic schemes for unsteady compressible flow simulations, VKI for Fluid Dynamics Lecture Series 1998-03, Computational Fluid Dynamics, 1998.
[26] 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.
[27] K. Xu, M.L. Mao, A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow, J. Comput. Phys. 203 (2005) 405–421.
[28] K. Xu, X. He, Lattice Boltzmann method and gas-kinetic BGK scheme in the low-Mach number viscous flow simulations, J. Comput. Phys. 190 (2003) 100–117.