

# A Robust and Accurate LED-BGK Solver on Unstructured Adaptive Meshes

Chongam Kim\* and Antony Jameson†

\**Department of Aerospace Engineering, Seoul National University, Seoul, Korea; and* †*Department of Aeronautics and Astronautics, Stanford University, Stanford, California 94305-4035*

Received July 29, 1997; revised March 10, 1998

---

Starting from the BGK model of the Boltzmann equation, we develop a robust and accurate finite volume gas kinetic scheme on unstructured triangular meshes. The proposed numerical approach is composed of two steps—an initial reconstruction step and a gas evolution step. In the initial reconstruction step, an unstructured version of the local extremum diminishing interpolation is applied to the conservative variables and to compute left and right states along a node edge. In the gas evolution step, the local integral solution of the BGK model is used to compute numerical fluxes at a cell interface. This approach provides an alternative to Riemann solvers and yields numerical schemes which possess many desirable properties that may not be found in Godunov-type schemes. A classic  $h$ -refinement adaptive procedure is implemented to increase the spatial resolution of high-speed unsteady flow characteristics such as shock waves, contact discontinuities, or expansion waves with minimal computational costs and memory overheads. This procedure involves mesh enrichment/coarsening steps to either insert nodes on an edge center in high-gradient regions or delete nodes in over-resolved regions. Numerical results of several test cases for unsteady compressible inviscid flows are presented. In order to verify the accuracy and robustness of the current numerical approach, the computed results are compared with analytical solutions, experimental data, the results of structured mesh calculations, and the results obtained by widely used flux splitting methods. © 1998 Academic Press

*Key Words:* LED (local extremum diminishing) interpolation; BGK-based gas kinetic scheme; unstructured mesh-based flow solver;  $h$ -refinement mesh adaptation.

---

## 1. INTRODUCTION

Since computational methods were first introduced as a way to simulate, analyze, and predict fluid dynamics problems, considerable progress has been made in the areas of compressible flow solvers and solution strategies [15]. At the same time, we have witnessed

tremendous advances in computer storage, speed, and architecture. These two factors have made possible the routine calculation of steady compressible flows over three-dimensional complex bodies. In spite of these advances, accurate computations of unsteady compressible flows over complex geometry still present a challenge. This is partly due to the complex physical interactions between linear and non-linear waves. As unsteady computations of complex flows are becoming an essential tool for practical application and fundamental research, there is a need to design a low-dissipative, robust numerical scheme on unstructured triangular meshes. Quite often, the requirements of robustness and accuracy in the design of a numerical scheme are in conflict with each other: If a scheme A is robust, it is unnecessarily diffusive and, if a scheme B is accurate, it loses robustness. This suggests that intelligent control of numerical dissipation based on the physics of the flow may lead to the successful development of schemes which are both robust and accurate.

A high-resolution scheme can be examined in terms of an initial reconstruction and a gas evolution procedure. Higher order (more than second order) interpolation on an unstructured mesh is much more involved than on a structured mesh since neighboring cell vertices are not, in general, colinear. Additionally, all interpolated values should be reduced to first order accuracy to exclude unphysical oscillations across local extrema. Thus it seems reasonable to keep second order accuracy in both spatial and temporal discretization, and introduce a local mesh refinement step to preserve the required accuracy. Recently, Jameson developed the theory of non-oscillatory positive schemes in terms of the local extremum diminishing (LED) criteria [16]. Flux limited dissipation schemes based on the LED criteria have been implemented and validated for both inviscid and viscous flow computations on a structured mesh [18, 37]. It has been observed that the LED interpolation, unlike the TVD (total variation diminishing) interpolation, can be extended to an unstructured mesh while maintaining the positive coefficients of the discrete form in a scalar conservation law. This can be carried out by calculating gradients of appropriate neighboring triangles or edges and applying a monotonic limiter.

Riemann solvers, approximate or exact, have gained much acceptance as a way to describe the gas evolution process [10, 13, 35]. In most test cases, Riemann solvers produce good results. Nevertheless, it has been found that Riemann solvers can produce unphysical answers such as expansion shocks, carbuncle phenomena, odd-even decoupling, or failure of local linearization, etc. [9, 18, 32]. Although most of the problems can be cured by adding *ad hoc* fixes, the robustness of the original approach seems to be compromised as a result of incomplete remedies. After the introduction of a gas kinetic scheme based on the BGK model by Xu and Prendergast [40, 41], there has been continuous development in this area [19, 21, 42, 43]. Previous studies have unveiled interesting properties of BGK-based schemes. First, they satisfy the entropy and the positivity condition which are important in the computations of high-speed compressible flows. Second, they possess a multi-dimensional gas evolution character and the interpolation in the initial reconstruction step is quite flexible. Third, the BGK model produces the Navier–Stokes equation through the particle collision mechanism and describes a higher order gas evolution process. These properties cannot be found in most popular Riemann solvers and are generally regarded as limitations of the Riemann solver approach. Thus it is not surprising that BGK-based schemes can overcome many shortcomings of Riemann solvers and provide a good alternative to Riemann solvers for the gas evolution step.

BGK-based schemes are different from the class of schemes called Boltzmann-type schemes [8, 27, 30, 31]. The physical model for these schemes is the collisionless Boltzmann

equation, which does not include the interaction between left and right moving particles. The missing correlation corresponds to a gas kinetic Lax–Wendroff step [42] and in BGK-based schemes, it is coupled to kinetic flux vector splitting through the particle collision time [19]. In addition, Boltzmann-type schemes split the gas evolution step into a convective step and an instantaneous collision step, which makes the collision time and mean free path  $O(\Delta t), O(\Delta x)$ , respectively. This simplification is, however, unreasonable considering the characteristic time and length scale of molecular motion, and it inevitably leads to large numerical viscosity and heat conductivity. As a way to cure this problem, Moschetta and Pullin recently proposed a hybrid solver (EFMO) which augments the equilibrium flux method (EFM) with Osher's approximate Riemann solver in calculating fluxes for linearly degenerate subpath, i.e., contact discontinuity [29]. Unlike the hybrid approach, BGK-based schemes make local use of the full integral solution of the BGK model in a finite volume framework which allows us to compute a time-dependent gas distribution function at a cell interface and to obtain the numerical fluxes. This approach provides a physical model for particle collisions designed to reduce the large numerical viscosity inherently present in schemes based on the collisionless Boltzmann equation. BGK-based schemes also give Navier–Stokes solutions which follow directly from the BGK model, and the gas relaxation from an initial non-equilibrium to a final equilibrium state is associated with an increase of entropy.

In order to increase efficiently the resolution of physically important local phenomena, a mesh adaptation strategy is essential. Since unstructured triangular meshes facilitate mesh adaptation, an unstructured-mesh-based flow solver is very desirable. Most popular mesh adaptation strategies can be classified into two approaches: mesh movement ( $r$ -refinement) and mesh enrichment ( $h$ -refinement). The  $r$ -refinement method redistributes existing nodes at every adaptation, clustering them toward positions where the solution gradient is relatively large. An advantage of the  $r$ -refinement method is that the total number of nodes and the grid topology are conserved. Thus there is no computational overhead on a flow solver and the flow solver is amenable to parallelization. However, the possibility of edge crossing during nodes movement and the need for remeshing make this approach cumbersome. Additionally, an accurate estimate of solution error and conservation of mesh smoothness after adaptation become difficult [3]. On the other hand, the  $h$ -refinement method alters the number of nodal points by adding or deleting nodes within the computational domain. Since only specific regions of the computational domain are altered, the node density can be locally controlled and the regularity of the initial mesh can be maintained. Another advantage of the  $h$ -refinement method over the  $r$ -refinement method is that the error indicator only needs to be sensitive enough to trigger mesh enrichment when there is a rapid variation of the flow field. In the analysis of unsteady flow with mesh adaptation strategy, it is necessary not only to refine high-gradient regions but also to de-refine over-resolved regions. The adaptation routine therefore consists of mesh enrichment/coarsening steps.

In the present work, an unstructured version of the LED interpolation is employed in the initial reconstruction of the data. An advanced BGK-based scheme which has been designed by the authors [43] is used to obtain fluxes in the gas evolution step. Then an  $h$ -refinement adaptation routine is introduced to increase spatial resolution. In Section 2, the finite volume LED-BGK scheme is described in terms of the LED interpolation and a BGK flow solver. In Section 3, the mesh adaptation strategy adopted is described and the initial mesh generation method is briefly mentioned. Finally, in Section 4, numerical results for test cases which contain typical characteristics of high-speed unsteady flows are given.

These are compared with structured mesh results or other popular flux splitting methods, such as characteristic splitting using Roe averaging and CUSP (convective upwind split pressure) splitting by Jameson.

## 2. FINITE VOLUME LED-BGK SOLVER

In gas kinetic theory, it is assumed that the macroscopic fluid flow results from the collective motion of a large number of molecules. The complete description of a particle motion is assumed to be described by the governing equation of the particle distribution function  $f(\mathbf{x}, t, \mathbf{u}, \xi)$ , where  $\mathbf{x}$  is a space variable vector,  $t$  is a time variable,  $\mathbf{u}$  is a particle velocity vector in phase space, and  $\xi$  stands for the internal degrees of freedom. Thus the most important step in a finite volume gas kinetic scheme is to design the time-dependent gas distribution function  $f$  at a cell interface. All flow variables or numerical fluxes can be computed from the moments of  $f$  in phase space. The design of a distribution function comes from the interpolation of macroscopic flow variables at the cell interface. The interpolation technique can be applied to the conservative, characteristic, or primitive variables.

### 2.1. The LED Interpolation on Unstructured Meshes

At the beginning of each time step, cell-averaged mass, momentum, and energy are given. For a higher order scheme, interpolation techniques should be used to capture subcell structure. The most successful interpolation techniques developed so far are usually based on the TVD, ENO, and LED principles [11, 12, 16, 36]. Although most of these approaches have been developed mainly for structured meshes, they can be applied to unstructured meshes by extending the one-dimensional concept. A unique feature of the LED interpolation is that the semi-discrete form for a scalar conservation law satisfies the positivity condition regardless of mesh types [16]. In order to examine this, a two-dimensional scalar hyperbolic conservation law is considered:

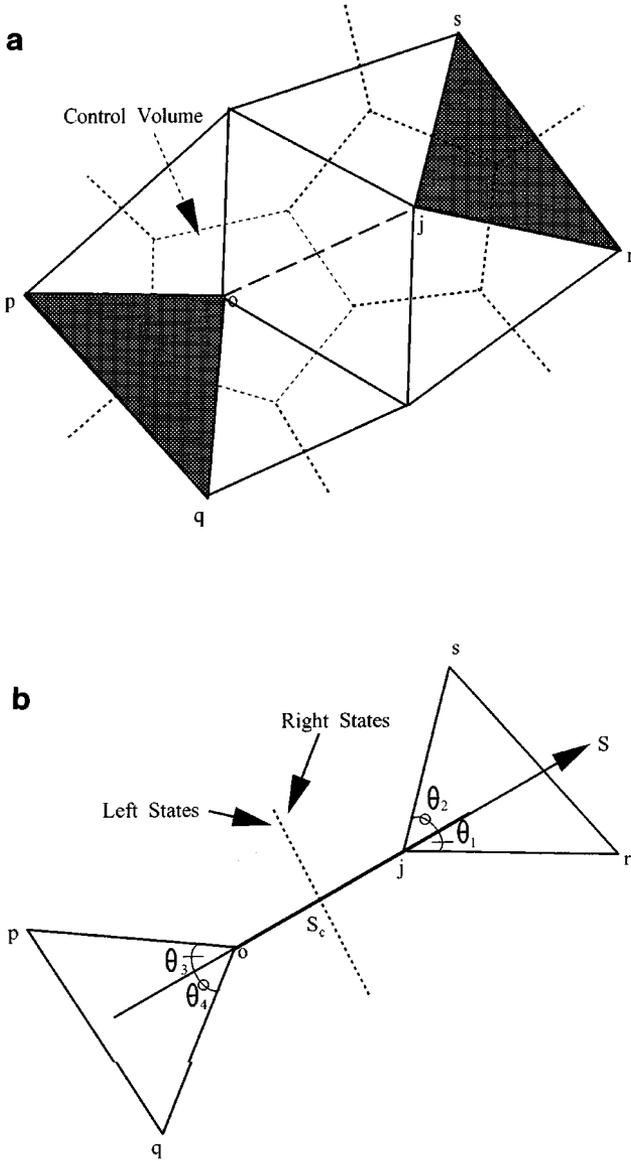
$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0.$$

For triangles as sketched in Fig. 1, flow variables are defined at cell vertex and the control volume is a centroid dual. By applying trapezoidal integration to the cell with vertex  $o$  (i.e., central differencing the flux integral), we obtain

$$S \frac{du_o}{dt} = - \sum_j (f_j \Delta y_j - g_j \Delta x_j), \tag{1}$$

where  $\Delta x_j = \frac{1}{2}(x_{j+1} - x_{j-1})$ ,  $\Delta y_j = \frac{1}{2}(y_{j+1} - y_{j-1})$ , and  $j$  covers all neighboring nodes connected to node  $o$ .  $S$  is the area of the cell. If we note that  $\sum_j \Delta x_j = \sum_j \Delta y_j = 0$ , Eq. (1) can be changed into the form which sums up the flux along neighboring edge  $jo$ :

$$S \frac{du_o}{dt} = - \sum_j \Delta \mathcal{F}_{jo} = - \sum_j [(f_j - f_o) \Delta y_j - (g_j - g_o) \Delta x_j]. \tag{2}$$



**FIG. 1.** Interpolation on unstructured meshes. (a) Control volume, edge ( $l_{jo}$ ) and neighboring triangles. (b) Local coordinates ( $S$ ).

This formulation, which was originally proposed in [14], is quite efficient with an edge-based data structure. After defining the local wave speed  $a_{jo} = \Delta \mathcal{F}_{jo} / \Delta u_{jo}$  along the edge  $jo$  and augmenting  $\Delta \mathcal{F}_{jo}$  with appropriate diffusive flux  $d_{jo}$ , we obtain

$$S \frac{du_o}{dt} = \sum_j c_{jo} \Delta u_{jo} = \sum_j (\alpha_{jo} - a_{jo}) \Delta u_{jo}, \quad (3)$$

with  $d_{jo} = \alpha_{jo} \Delta u_{jo}$  and  $\Delta u_{jo} = u_j - u_o$ . A limited diffusive flux of  $d_{jo}$  can be obtained by estimating neighboring differences. Let  $l_{jo}$  be the vector along the edge  $jo$ . Then

neighboring slopes are calculated by searching appropriate triangles as

$$\begin{aligned} \Delta^+ u_{jo} &= l_{jo} \cdot \nabla^+ u = \epsilon_{js}(u_s - u_j) + \epsilon_{jr}(u_r - u_j), \\ \Delta^- u_{jo} &= l_{jo} \cdot \nabla^- u = \epsilon_{op}(u_o - u_p) + \epsilon_{oq}(u_o - u_q), \end{aligned} \quad (4)$$

where  $\nabla^\pm$  are the gradients of  $u$  calculated at the neighboring triangles  $\Delta jsr$  and  $\Delta opq$  as shown in Fig. 1a. The coefficients  $\epsilon_{js}, \epsilon_{jr}, \epsilon_{op}, \epsilon_{oq}$  are all non-negative, since

$$\begin{aligned} \epsilon_{js} &= \frac{l_{oj}l_{jr}}{2A_{jsr}} \sin(\theta_1), & \epsilon_{jr} &= \frac{l_{oj}l_{js}}{2A_{jsr}} \sin(\theta_2), \\ \epsilon_{op} &= \frac{l_{oj}l_{oq}}{2A_{opq}} \sin(\theta_4), & \epsilon_{oq} &= \frac{l_{oj}l_{op}}{2A_{opq}} \sin(\theta_3). \end{aligned} \quad (5)$$

Here,  $l_{ab}$  is the length of the edge connecting nodes  $a, b$ ;  $A_{abc}$  is the area of the  $\Delta abc$ ; and  $\theta_i$  is the angle between edges as shown in Fig. 1b. When an edge of neighboring triangles is almost colinear to the edge  $jo$ , the choice of an appropriate neighboring triangle may not be unique. In this case, instead of evaluating the gradient, we choose the slope of the neighboring edge as the neighboring slope. Now, the limited diffusive flux can be defined as

$$d_{jo} = \alpha_{jo}[\Delta u_{jo} - L(\Delta^+ u_{jo}, \Delta^- u_{jo})]. \quad (6)$$

With this form, Eq. (3) becomes

$$S \frac{du_o}{dt} = \sum_j c_{jo} \Delta u_{jo} = \sum_j [(\alpha_{jo} - a_{jo}) \Delta u_{jo} + \phi(r_{jo}^-) \epsilon_{op} \Delta u_{po} + \phi(r_{jo}^-) \epsilon_{oq} \Delta u_{qo}], \quad (7)$$

where

$$r_{jo}^- = \frac{\Delta^+ u_{jo}}{\Delta^- u_{jo}}, \quad \phi(r) = L(r, 1) = L(1, r).$$

$L(u, v)$  is a limited average which satisfies the following properties:

- P1.  $L(u, v) = L(v, u)$
- P2.  $L(\alpha u, \alpha v) = \alpha L(u, v)$
- P3.  $L(u, u) = u$
- P4.  $L(u, v) = 0$  if  $u$  and  $v$  has an opposite sign; otherwise  $L(u, v)$  has the same sign as  $u$  and  $v$ .

Standard limiters can be incorporated into  $L$  operators. In the present work,  $\alpha_2$  limiter of  $L(u, v) = S(u, v) \text{Min}(\frac{|u+v|}{2}, \alpha|u|, \alpha|v|)$  with  $\alpha = 2$  is used, where  $S(u, v) = \frac{1}{2}(\text{sign}(u) + \text{sign}(v))$ . Since  $\phi(r)$  and the coefficients  $\epsilon$  of neighboring slopes are all non-negative, each coefficient  $c_{jo}$  in Eq. (7) gives a positive contribution with  $\alpha_{jo} \geq |a_{jo}|$ . Then from Eq. (7), if  $u_o$  is a local maximum,  $\Delta u_{ko} \leq 0$  with  $k = j, p, q$  and  $du_o/dt \leq 0$ . If  $u_o$  is a local minimum,  $\Delta u_{ko} \geq 0$  and  $du_o/dt \geq 0$ . Thus the LED criteria is satisfied. In the case of systems of equations, the left and right states across the cell interface  $s = s_c$  are estimated as

$$u^l = u_o + 0.5L(\Delta^+ u_{jo}, \Delta^- u_{jo}), \quad u^r = u_j - 0.5L(\Delta^+ u_{jo}, \Delta^- u_{jo}),$$

or the limited diffusive flux of Eq. (6) can be directly applied to either characteristic or conservative variables.

## 2.2. A BGK Flow Solver

In order to avoid mathematical complexity, we present a two-dimensional BGK flow solver. Since a single scalar distribution function  $f$  contains all information about the macroscopic flow variables as well as their transport coefficients, the BGK solver in three dimensions can be derived in a similar way. The BGK model in two dimensions can be written as

$$f_t + uf_x + vf_y = \frac{g - f}{\tau}, \quad (8)$$

where  $f$  is the real gas distribution function and  $g$  is the equilibrium state approached by  $f$  over a collision time scale  $\tau$ . Both  $f$  and  $g$  are functions of space  $(x, y)$ , time  $t$ , particle velocity  $(u, v)$ , and  $\xi$ .  $\xi = (\xi_1, \xi_2, \dots, \xi_K)$  is the  $K$ -dimensional vector of internal velocities with  $\xi^2 \equiv \sum_{i=1}^K \xi_i^2$ . It is convenient to introduce internal velocities to properly describe the internal energy and specific heats for a perfect gas. For instance, in a  $D$ -dimensional flow, the molecular velocity components normal to the macroscopic flow directions are added to the internal degrees of freedom. Then for  $\gamma = \frac{n+2}{n}$  with  $n = K + D$ , where  $n$  is the effective total number of degrees of freedom of a gas,  $K$  is given by  $-D + \frac{2}{\gamma-1}$ . Thus for the two-dimensional case with  $D = 2$ ,  $K$  is equal to  $\frac{4-2\gamma}{\gamma-1}$ . The particle collision time  $\tau$  depends on the local macroscopic flow variables, such as temperature and density. The equilibrium state is usually assumed to be a Maxwellian, with the formulation

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

where  $\rho$  is the density, and  $(U, V)$  are the macroscopic flow velocities in  $(x, y)$  directions. The relations between mass  $\rho$ , momentum  $(P, Q)$ , and energy densities  $\epsilon$  with the distribution function  $f$  are

$$\begin{pmatrix} \rho \\ P \\ Q \\ \epsilon \end{pmatrix} = \int f \psi d\Xi, \quad (9)$$

where  $\psi$  is the vector of moments

$$\psi = \left[ 1, u, v, \frac{(u^2 + v^2 + \xi^2)}{2} \right]^T,$$

and  $d\Xi = du dv d\xi$  is the volume element in phase space. Since mass, momentum, and energy are conserved during particle collisions,  $f$  and  $g$  must satisfy the conservation constraint of

$$\int (g - f) \psi d\Xi = 0, \quad (10)$$

at all  $(x, y)$  and  $t$ .

Here, the relation between the BGK model and governing equations of fluid dynamics is examined. Using the notation

$$\langle a, b \rangle = \int ab \, d\Xi, \quad \langle a, b \rangle^+ = \int_{u>0} ab \, d\Xi, \quad \langle a, b \rangle^- = \int_{u<0} ab \, d\Xi$$

and Eqs. (8) and (10), we obtain

$$\langle \psi, f \rangle_t + \langle u\psi, f \rangle_x + \langle v\psi, f \rangle_y = \frac{\langle \psi, g - f \rangle}{\tau} = 0. \quad (11)$$

Using Eq. (10) again, the final moment equation of the BGK model is

$$\langle \psi, g \rangle_t + \langle u\psi, g \rangle_x + \langle v\psi, g \rangle_y = \langle u\psi, g - f \rangle_x + \langle v\psi, g - f \rangle_y. \quad (12)$$

In a smooth region, it is reasonable to assume that the flow is in a local thermodynamic equilibrium state. For the equilibrium state with  $f = g$ , Eq. (12) becomes the same as the moment equations of the collisionless Boltzmann equation and the Euler equations can be obtained,

$$\langle \psi, g \rangle_t + \langle u\psi, g \rangle_x + \langle v\psi, g \rangle_y = \int (g_t + ug_x + vg_y) \psi \, d\Xi = 0,$$

and the corresponding Euler equations are

$$\begin{aligned} & \left( \begin{array}{c} \rho \\ \rho U \\ \rho V \\ \frac{\rho}{2}(U^2 + V^2 + \frac{K+2}{2\lambda}) \end{array} \right)_t + \left( \begin{array}{c} \rho U \\ \rho U^2 + \frac{\rho}{2\lambda} \\ \rho UV \\ \frac{\rho U}{2}(U^2 + V^2 + \frac{K+4}{2\lambda}) \end{array} \right)_x \\ & + \left( \begin{array}{c} \rho V \\ \rho UV \\ \rho V^2 + \frac{\rho}{2\lambda} \\ \frac{\rho V}{2}(U^2 + V^2 + \frac{K+4}{2\lambda}) \end{array} \right)_y = 0, \end{aligned}$$

where the pressure term is  $p = \rho/2\lambda$ . The recurrence formulas for all the moment calculations are given in the Appendix.

In a physically dissipative region, the flow is in a non-equilibrium state. The behavior of  $f$  in this region can be quantitatively examined by the Chapman–Enskog expansion [22]. Up to the first order of  $\tau$ , the departure of  $f$  from an equilibrium state is  $f = g - \tau(g_t + ug_x + vg_y)$ . With this  $f$ , Eq. (8) becomes

$$\begin{aligned} & \langle \psi, g \rangle_t + \langle u\psi, g \rangle_x + \langle v\psi, g \rangle_y \\ & = \tau \langle \psi, ug_{xt} + vg_{yt} + 2uv g_{xy} + u^2 g_{xx} + v^2 g_{yy} \rangle + O(\tau^2) \end{aligned}$$

or

$$\int (g_t + ug_x + vg_y) \psi \, d\Xi = \tau \int (ug_{xt} + vg_{yt} + 2uv g_{xy} + u^2 g_{xx} + v^2 g_{yy}) \psi \, d\Xi + O(\tau^2).$$

After all the moments, including time derivative terms, have been integrated out, the Navier–Stokes equations are obtained as

$$\begin{aligned}
 & \left( \begin{array}{c} \rho \\ \rho U \\ \rho V \\ \frac{\rho}{2}(U^2 + V^2 + \frac{K+2}{2\lambda}) \end{array} \right)_t + \left( \begin{array}{c} \rho U \\ \rho U^2 + \frac{\rho}{2\lambda} \\ \rho UV \\ \frac{\rho U}{2}(U^2 + V^2 + \frac{K+4}{2\lambda}) \end{array} \right)_x + \left( \begin{array}{c} \rho V \\ \rho UV \\ \rho V^2 + \frac{\rho}{2\lambda} \\ \frac{\rho V}{2}(U^2 + V^2 + \frac{K+4}{2\lambda}) \end{array} \right)_y \\
 & = \tau \left( \begin{array}{c} 0 \\ C_1((K+1)U_x - V_y) \\ C_2(U_y + V_x) \\ C_1U((K+1)U_x - V_y) + C_2V(U_y + V_x) + C_3(\frac{1}{\lambda})_x \end{array} \right)_x \\
 & + \tau \left( \begin{array}{c} 0 \\ C_2(U_y + V_x) \\ C_1((K+1)V_y - U_x) \\ C_2U(U_y + V_x) + C_1V((K+1)V_y - U_x) + C_3(\frac{1}{\lambda})_y \end{array} \right)_y, \quad (13)
 \end{aligned}$$

where  $C_1 = \frac{\rho}{(K+2)\lambda}$ ,  $C_2 = \frac{\rho}{2\lambda}$ , and  $C_3 = \frac{(K+4)\rho}{8\lambda}$ . The viscosity coefficient is  $\mu = \tau\rho$ , the heat conductivity coefficient is  $\kappa = \mu c_p$ , and  $c_p$  is the specific heat constant.

After introducing a local coordinate in terms of  $s$  (see Fig. 1), the general solution of  $f$  at the cell interface  $s = s_c$  and time  $t$  is [22]

$$f(s_c, t, u, v, \xi) = \frac{1}{\tau} \int_0^t g(s', t', u, v, \xi) e^{-(t-t')/\tau} dt' + e^{-t/\tau} f_0(s_c - ut - vt), \quad (14)$$

where  $s' = s_c - u(t - t') - v(t - t')$  is the trajectory of a particle and  $f_0$  is the initial non-equilibrium distribution function  $f$  at the beginning of each time step.  $(u, v)$  are particle velocities normal and tangential to a cell boundary. From Eq. (14), we may notice that  $g$  and  $f_0$  must be specified in order to obtain  $f$ .

In the present work,  $f_0$  and  $g$  around the cell interface are assumed to be

$$f_0 = \begin{cases} g^l [1 + a^l (s - s_c)], & s \leq s_c \\ g^r [1 + a^r (s - s_c)], & s \geq s_c \end{cases} \quad (15)$$

and

$$g = \begin{cases} g_0 [1 + \bar{a}^l (s - s_c) + At], & s \leq s_c \\ g_0 [1 + \bar{a}^r (s - s_c) + At], & s \geq s_c \end{cases} \quad (16)$$

where  $g^l$ ,  $g^r$ , and  $g_0$  are local Maxwellian distribution functions to be determined, which are located, respectively, to the left, to the right, and in the middle of a cell interface.  $a^l$ ,  $a^r$ ,  $\bar{a}^l$ ,  $\bar{a}^r$ ,  $A$  are local spatial and time slopes. In the expansion of  $f_0$ , two local Maxwellians from the left and right side of cell interface are used to capture a physical discontinuity properly. For

example, since the upstream and downstream gas distribution functions are usually different across a shock wave, a splitting of  $f_0$  is needed to reflect this situation. In the expansion of  $g$ , the possibility of discontinuous slopes has been allowed. This increases substantially the robustness of the present scheme without sacrificing accuracy. The dependence of slopes ( $m = a, \bar{a}, A$ ) on the particle velocities can be obtained from the Taylor expansion of a Maxwellian and has the form

$$m = m_1^* + m_2^*u + m_3^*v + m_4^*\frac{(u^2 + v^2 + \xi^2)}{2} = m^{*T} \psi,$$

where all components of the slope vector  $m^* = (m_1^*, m_2^*, m_3^*, m_4^*)^T$  are local constants and  $*$  stands for the left ( $l$ ) or right ( $r$ ) state. Thus the actual expression for the spatial and time slopes is

$$a^l = a^{lT} \psi, \quad a^r = a^{rT} \psi, \quad \bar{a}^l = \bar{a}^{lT} \psi, \quad \bar{a}^r = \bar{a}^{rT} \psi, \quad A = A^T \psi.$$

In the LED interpolation described in Section 2.1, we have obtained the left and right side pointwise values of  $(\rho^l, P^l, \dots, Q^r, \epsilon^r)$ . By the relation between the gas distribution function  $f_0$  and the macroscopic variables (Eq. (9)), we obtain

$$\int g^l \psi d\Xi = \begin{pmatrix} \rho^l \\ P^l \\ Q^l \\ \epsilon^l \end{pmatrix}; \quad \int g^l a^l \psi d\Xi = \begin{pmatrix} \frac{\rho^l - \rho_o}{\Delta s^-} \\ \frac{P^l - P_o}{\Delta s^-} \\ \frac{Q^l - Q_o}{\Delta s^-} \\ \frac{\epsilon^l - \epsilon_o}{\Delta s^-} \end{pmatrix}$$

and

$$\int g^r \psi d\Xi = \begin{pmatrix} \rho^r \\ P^r \\ Q^r \\ \epsilon^r \end{pmatrix}; \quad \int g^r a^r \psi d\Xi = \begin{pmatrix} \frac{\rho_j - \rho^r}{\Delta s^+} \\ \frac{P_j - P^r}{\Delta s^+} \\ \frac{Q_j - Q^r}{\Delta s^+} \\ \frac{\epsilon_j - \epsilon^r}{\Delta s^+} \end{pmatrix}, \tag{17}$$

where  $\Delta s^- = s_c - s_o$  and  $\Delta s^+ = s_j - s_c$  (see Fig. 1). With the definition of the Maxwellian distributions for the left and right states

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

and from Eq. (17), all the parameters in  $g^*$  can be uniquely determined from

$$\begin{pmatrix} \rho^* \\ U^* \\ V^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} \rho^* \\ P^*/\rho^* \\ Q^*/\rho^* \\ \frac{0.25(K+2)\rho^*}{(\epsilon^* - (P^{*2} + Q^{*2})/2\rho^*)} \end{pmatrix}. \tag{18}$$

All the quantities on the right-hand side of Eq. (18) are obtained from the LED interpolation. Once  $g^*$  is obtained from Eq. (18), the slope of  $a^*$  can be computed. For example, for the right state, we obtain

$$\begin{pmatrix} \frac{\rho_j - \rho^r}{\rho^r \Delta s^+} \\ \frac{P_j - P^r}{\rho^r \Delta s^+} \\ \frac{Q_j - Q^r}{\rho^r \Delta s^+} \\ \frac{\epsilon_j - \epsilon^r}{\rho^r \Delta s^+} \end{pmatrix} = \frac{1}{\rho^r} \int g^r a^r \psi d\Xi = M^r a^r, \quad (19)$$

where the matrix  $M^r = [M_{ij}^r] = 1/\rho^r \int g^r \psi \psi^T d\Xi$  is

$$M^r = \begin{pmatrix} 1 & U^r & V^r & M_1 \\ U^r & U^{r2} + \frac{1}{2\lambda^r} & U^r V^r & M_2 \\ V^r & U^r V^r & V^{r2} + \frac{1}{2\lambda^r} & M_3 \\ M_1 & M_2 & M_3 & M_4 \end{pmatrix},$$

where

$$\begin{aligned} M_1 &= \frac{1}{2} \left( U^{r2} + V^{r2} + \frac{K+2}{2\lambda^r} \right), \\ M_2 &= \frac{1}{2} \left( U^{r3} + V^{r2} U^r + \frac{(K+4)U^r}{2\lambda^r} \right), \\ M_3 &= \frac{1}{2} \left( V^{r3} + U^{r2} V^r + \frac{(K+4)V^r}{2\lambda^r} \right), \\ M_4 &= \frac{1}{4} \left[ (U^{r2} + V^{r2})^2 + \frac{(K+4)(U^{r2} + V^{r2})}{\lambda^r} + \frac{(K^2 + 6K + 8)}{4\lambda^{r2}} \right]. \end{aligned}$$

From Eq. (19), the components of slope vector  $a^r = (a_1^r, a_2^r, a_3^r, a_4^r)^T$  can be efficiently obtained by inverting a symmetric matrix  $M^r$  explicitly. Since the matrix  $M^l = [M_{ij}^l] = 1/\rho^l \int g^l \psi \psi^T d\Xi$  has the same structure of  $M^r$ ,  $a^l = (a_1^l, a_2^l, a_3^l, a_4^l)^T$  can be obtained using a similar procedure.

By determining  $f_0$ , the corresponding values of  $\rho_0$ ,  $U_0$ ,  $V_0$ , and  $\lambda_0$  in  $g_0$  with

$$g_0 = \rho_0 \left( \frac{\lambda_0}{\pi} \right)^{(K+2)/2} e^{-\lambda_0[(u-U_0)^2 + (v-V_0)^2 + \xi^2]}$$

can be determined as follows. If we take the limit  $t \rightarrow 0$  in Eq. (14) and substitute its solution into Eq. (10), the conservation constraint at  $(s = s_c, t = 0)$  gives

$$\int g_0 \psi d\Xi = \int_{u>0} g^l \psi d\Xi + \int_{u<0} g^r \psi d\Xi. \quad (20)$$

Since  $\lambda_0$  can be found from  $\rho_0$ ,  $U_0$ ,  $V_0$ , and  $\epsilon_0$  through the relation

$$\lambda_0 = 0.25(K+2)\rho_0 \left/ \left[ \epsilon_0 - \frac{\rho_0(U_0^2 + V_0^2)}{2} \right] \right.,$$

we only need to know  $(\rho_0, U_0, V_0, \epsilon_0)$ , which can be expressed as moments of  $g^l$  and  $g^r$ . From Eq. (20), we obtain

$$\begin{pmatrix} \rho_0 \\ P_0 \\ Q_0 \\ \epsilon_0 \end{pmatrix} = \begin{pmatrix} \rho^l \langle 1, g^l \rangle^+ + \rho^r \langle 1, g^r \rangle^- \\ \rho^l \langle u, g^l \rangle^+ + \rho^r \langle u, g^r \rangle^- \\ \rho^l \langle v, g^l \rangle^+ + \rho^r \langle v, g^r \rangle^- \\ \rho^l \langle \mathbf{v}^2, g^l \rangle^+ / 2 + \rho^r \langle \mathbf{v}^2, g^r \rangle^- / 2 \end{pmatrix}, \quad (21)$$

where  $\mathbf{v}^2 = u^2 + v^2 + \xi^2$ . Then  $\bar{a}^l$  and  $\bar{a}^r$  of  $g$  in Eq. (16) can be obtained through the relation

$$\begin{pmatrix} \frac{\rho_0 - \rho_0}{\rho_0 \Delta s^-} \\ \frac{P_0 - P_0}{\rho_0 \Delta s^-} \\ \frac{Q_0 - Q_0}{\rho_0 \Delta s^-} \\ \frac{\epsilon_0 - \epsilon_0}{\rho_0 \Delta s^-} \end{pmatrix} = \bar{M}^0 \bar{a}^l, \quad \begin{pmatrix} \frac{\rho_j - \rho_0}{\rho_0 \Delta s^+} \\ \frac{P_j - P_0}{\rho_0 \Delta s^+} \\ \frac{Q_j - Q_0}{\rho_0 \Delta s^+} \\ \frac{\epsilon_j - \epsilon_0}{\rho_0 \Delta s^+} \end{pmatrix} = \bar{M}^0 \bar{a}^r.$$

The matrix  $\bar{M}^0 = [\bar{M}_{ij}^0] = 1/\rho_0 \int g_0 \psi \psi^T d\Xi$  has the same structure of  $M^r$ . Thus  $\bar{a}^r = (\bar{a}_1^r, \bar{a}_2^r, \bar{a}_3^r, \bar{a}_4^r)^T$  and  $\bar{a}^l = (\bar{a}_1^l, \bar{a}_2^l, \bar{a}_3^l, \bar{a}_4^l)^T$  can be found by following the procedure in Eq. (19).

Up to this point, we have determined two half-Maxwellians ( $g^l, g^r$ ) for  $f_0$  and one whole Maxwellian distribution function ( $g_0$ ) at the cell interface, and they represent the non-equilibrium state  $f_0$  and equilibrium state  $g_0$ . All the slopes in the expression of  $a^l, a^r$  in  $f_0$  and  $\bar{a}^l, \bar{a}^r$  in  $g$  are obtained from the slopes of macroscopic variables. The construction of two slopes for  $g$  gives more freedom to describe complicated flow situations and the slopes of  $\bar{a}^l$  and  $\bar{a}^r$  represent the viscosity and heat conduction effects for Navier–Stokes solutions.

After we substitute Eqs. (15) and (16) into Eq. (14), the final gas distribution function at a cell interface is expressed as

$$\begin{aligned} f(s_c, t, u, v, \xi) = & (1 - e^{-t/\tau})g_0 + [\tau(-1 + e^{-t/\tau}) + te^{-t/\tau}](\bar{a}^l H(u) \\ & + \bar{a}^r (1 - H(u)))ug_0 + \tau(t/\tau - 1 + e^{-t/\tau})Ag_0 \\ & + e^{-t/\tau}[(1 - uta^l)H(u)g^l + (1 - uta^r)(1 - H(u))g^r], \quad (22) \end{aligned}$$

where  $H(u)$  is the unit step function. The only unknown term in Eq. (22) is the time evolution term  $A$ . Since both  $f$  (Eq. (22)) and  $g$  (Eq. (16)) contain  $A$ , we apply again the conservation constraint of Eq. (10) at  $s = s_c$  and integrate it over the time step  $T$  to get

$$\int_0^T \int (g - f) \psi d\Xi dt = 0,$$

which gives

$$\begin{aligned} \bar{M}^0 A = & \frac{1}{\rho_0} \int [\gamma_1 u (\bar{a}^l H(u) + \bar{a}^r (1 - H(u))g_0) \\ & + \gamma_2 u (a^l H(u)g^l + a^r (1 - H(u))g^r)] \psi d\Xi, \quad (23) \end{aligned}$$

where

$$\begin{aligned}\gamma_0 &= T - \tau(1 - e^{-T/\tau}), \\ \gamma_1 &= (-T + 2\tau(1 - e^{-T/\tau}) - Te^{-T/\tau})/\gamma_0, \\ \gamma_2 &= (Te^{-T/\tau} - \tau(1 - e^{-T/\tau}))/\gamma_0.\end{aligned}$$

All the moments of the Maxwellian on the right-hand side of Eq. (23) can be found in the Appendix and the above equation can be solved for  $A = (A_1, A_2, A_3, A_4)^T$  as in Eq. (19).

Finally, the numerical fluxes at the cell interface can be computed as

$$\begin{pmatrix} \mathcal{F}_\rho \\ \mathcal{F}_p \\ \mathcal{F}_Q \\ \mathcal{F}_\epsilon \end{pmatrix}_{s=s_c} = \int_0^T \int u f_{s_c} \psi d\Xi dt, \quad (24)$$

where  $f_{s_c} = f(s_c, t, u, v, \xi)$  is given in Eq. (22).

### 3. ADAPTATION STRATEGY

Any mesh adaptation scheme involves two essential steps: selection of adaptation criteria and a mesh refinement routine. Adaptation criteria detect appropriate regions which need further resolution. An optimal adaptation criterion would provide the error of the computed solution. Since this needs the information on the exact solution, approximate feature detection or error estimation techniques are commonly used. For example, Löhner *et al.* have successfully used an approximate  $L_2$ -norm error [23, 24]. Berger *et al.* have developed an error estimate based on the Richardson extrapolation on a rectangular mesh [5, 6]. Although a rigorous analysis on *a posteriori* error for a scalar conservation law has been carried out by Hansboro and Johnson in the context of the finite element method [17], an estimator in most cases is based on a feature detection technique due to its simplicity and efficiency. In the present work, a simple feature detection technique has been used to capture regions which need extra flowfield resolution—around a shock wave, a contact discontinuity, and expansion waves. For a given triangle, extrema and arithmetic average of a flow variable ( $q$ ) are calculated. Then the approximate magnitude of a flow gradient is measured by

$$\Delta_q = \text{Max} \left( \left| \frac{\bar{q} - q_{\max}}{\bar{q}} \right|, \left| \frac{\bar{q} - q_{\min}}{\bar{q}} \right| \right),$$

where  $q_{\max} = \max_i(q_i)$ ,  $q_{\min} = \min_i(q_i)$  and  $\bar{q} = \sum_{i=1}^N q_i / N$ . That is, the difference between extrema and the average is normalized by the average value. In practice, if  $\Delta_q > \epsilon_q$  for a threshold value  $\epsilon_q$ , the triangle is flagged for adaptation.

Since the mesh refinement routine is based on the solution obtained on a given mesh, the mesh should be generated to maximize mesh quality. For example, if the initial mesh is too coarse or irregular, the computed results would be unable to resolve or detect certain flow characteristics, giving a solution too poor to be adapted. The initial mesh generation method adopted in this paper is based on Rebay's point placement algorithm combined with the

Delaunay triangulation [34]. Starting from the given boundary points, the initial triangulation is carried out using the Delaunay criteria. Since that mesh is usually unacceptable in term of regularity, an additional point is introduced successively in a position which can produce a regular mesh as a result of the Delaunay triangulation. This routine is iterated until all triangles generated satisfy certain regularity requirements. See [2, 34] for the details.

Based on the solution obtained on the given mesh, the triangles to be adapted are determined according to adaptation criteria. The adaptation level which indicates the degree of adaptation is initialized as zero for all triangles. Adapted triangles are typically divided into four subtriangles by inserting nodes on edge centers and the adaptation level is increased by one. The neighboring triangles which share an edge with adapted triangles are flagged as boundary triangles. They are divided into two subtriangles and the corresponding adaptation level is increased by half. If a boundary triangle shares more than two edges with adapted triangles, it also should be flagged to be adapted. As the level of adaptation increases, however, several other cases are possible, as shown in Fig. 2. This method seems

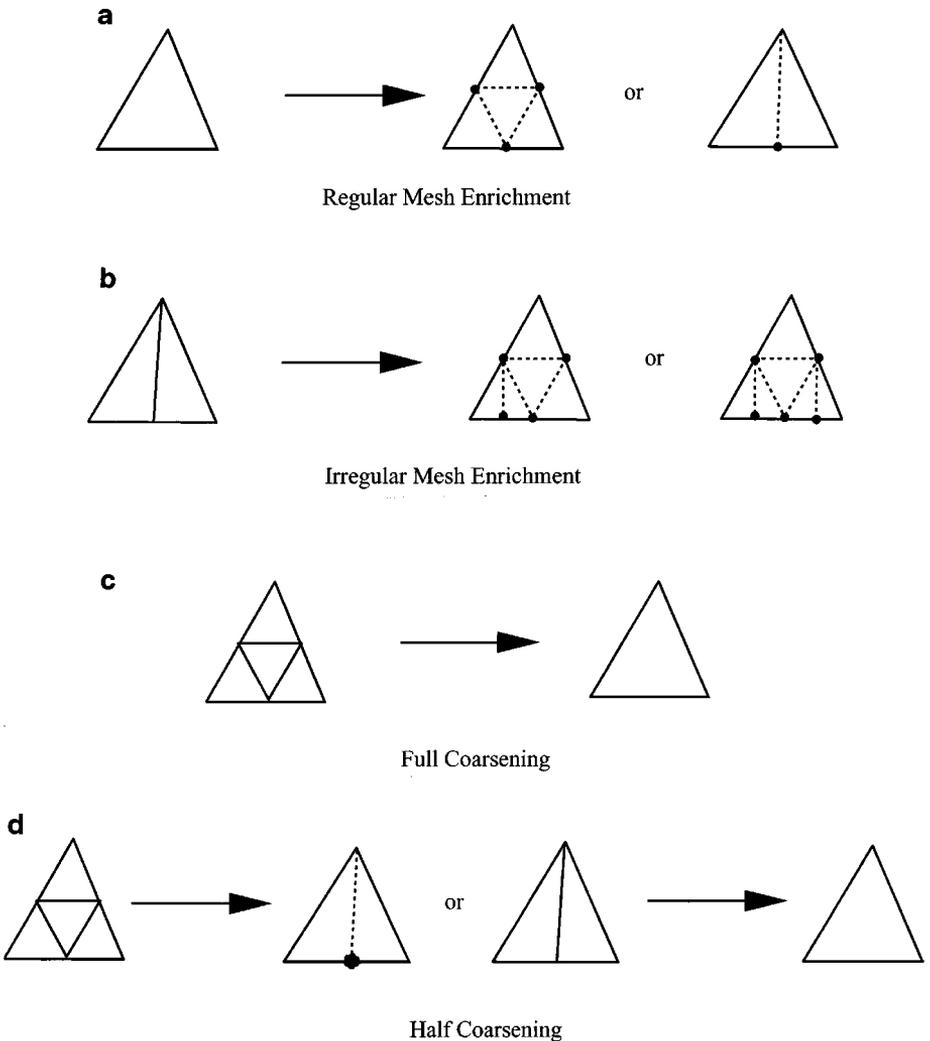


FIG. 2. Mesh adaptation steps. (a), (b) Mesh enrichment. (c), (d) Mesh coarsening.

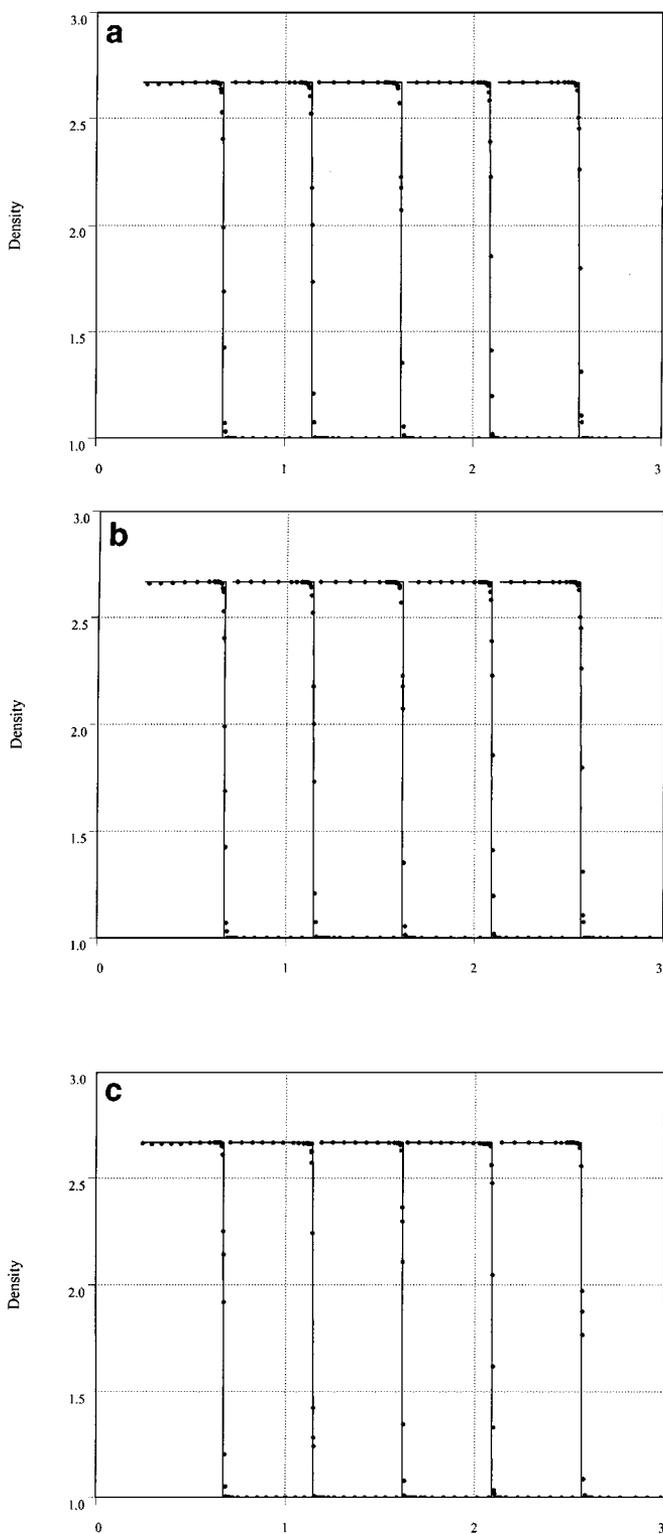


on the wall boundary. Flow variables at dummy nodes  $E$  and  $F(q_{E,F})$  are defined such that  $q_{E,F} = -q_{C,B}$  for normal velocity and  $q_{E,F} = q_{C,B}$  for other variables. Then the left and right states across cell boundary ( $ab, bc, cd, de$ ) can be estimated using one-dimensional LED interpolation technique. Flow variables at one side of cell boundary  $Oa, Oe$  are calculated by linear interpolation, and the values of the other side are defined as in the case of the other boundary. Once all interpolated states are determined, fluxes across the cell interface can be evaluated by exactly following the procedure described in Section 2.2.

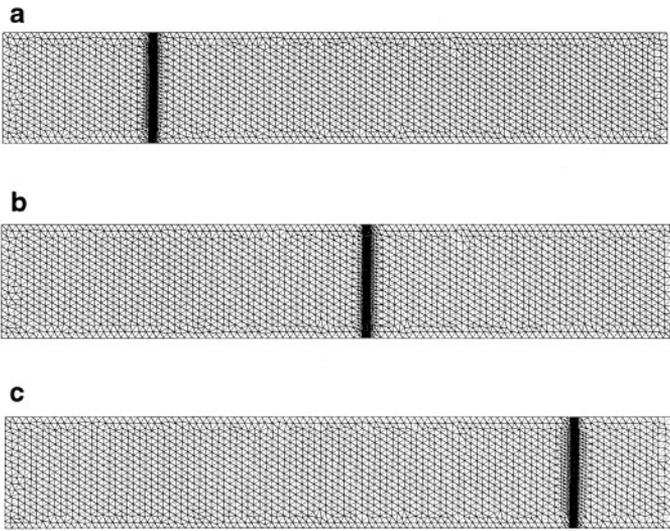
The first test case treats a moving shock with a speed  $M_s = 1.5$  that is propagated in a two-dimensional duct of  $(x, y) = (1.2, 0.2)$ . Although this is a relatively simple test, it provides a good way to check shock capturing properties of LED schemes augmented with mesh adaptation. Figure 4 shows the density profile along  $y = 0.1$  line at five different times. All schemes yield monotonic shock profiles and show an excellent agreement with exact solutions. As can be seen from Fig. 5, the number of nodes remains almost uniform during the entire adaptation process.

The second test case is the forward facing step with free stream Mach number 3. Since this test shows many important phenomena which typically happen in unsteady supersonic flows, it has been widely computed, especially on structured meshes [39]. In order to resolve the behavior of the expansion fan around the step corner, no special treatment such as an entropy fix was required. Figure 6 shows a typical initial mesh distribution. We can see the mesh regularity including the boundary region. The actual initial mesh contained 4836 nodes. Figure 7 shows the comparison of the density distribution at  $t = 4$ . An unstructured mesh without adaptation gives very competitive results with the structured mesh calculation. There are 16,140 cells in the structured mesh and 18,110 nodes in the unstructured mesh, respectively. Figure 8 shows the effect of adaptation at  $t = 3$ . Starting from the coarser mesh (4836 nodes), a typical final adapted mesh (LED-Csplit: 16,961; LED-CUSP: 16,016; LED-BGK: 17,649 nodes) shown in Fig. 8a is nicely clustered around shock waves, a contact discontinuity, and expansion regions. The corresponding computed results resolve local phenomena very clearly while keeping monotonic profiles. Especially, the inviscid instability—the so-called Kelvin–Helmholtz instability—of the contact discontinuity emerging from the shock triple point can be seen most clearly in the result of the LED-BGK scheme. In the case of the LED-Csplit scheme, the contact discontinuity is relatively diffuse and a noticeable length of Mach step has been observed. The reason is that if the same  $\alpha_2$  limiter is used as in other LED schemes, the adapted mesh around the step corner produces an overexpansion which leads to negative density and/or pressure. A relatively diffusive limiter such as the Minmod limiter had to be used to prevent that. This indicates the importance of robustness in designing a numerical scheme.

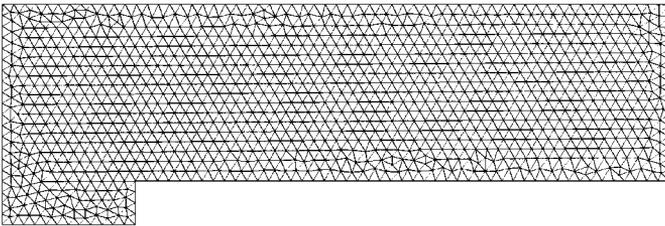
The third test case is the double Mach reflection with a shock speed  $M_s = 10.0$ . Initially a shock wave with  $M_s = 10.0$  is located at the entrance of a  $30^\circ$  wedge and propagated along the wedge. The computational domain is a rectangle of length  $(x, y) = (4, 1)$  and the entrance of the wedge is located at  $(x, y) = (1/6, 0)$ . The initial mesh for this case contained 16,986 nodes and the final adapted mesh contained 36,245 nodes. Figure 9b shows the density distribution of the computed results at  $t = 0.25$ . Compared with the results of the calculation on a rectangular mesh ( $360 \times 120$  cells, Fig. 9a), the computation without adaptation gives a similar result. The number of nodes used is about one-third the number of nodes in the rectangular mesh. Figures 9b and 9c show the density comparison between the uniform mesh and the adapted mesh. The adapted mesh is a result of two-level adaptation on the uniform mesh. Again, the resolution of shock waves is greatly improved. The resolution



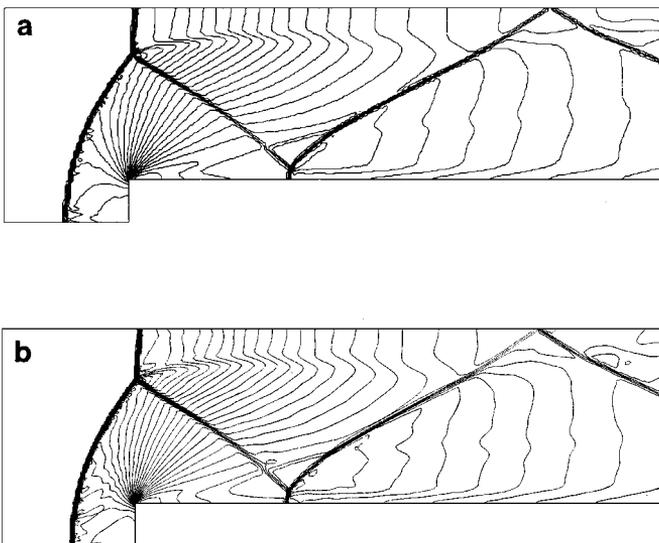
**FIG. 4.** Two-dimensional moving shock with  $M_s = 1.5$ . Computed results ( $\cdots$ ) are compared with exact solutions ( $\text{---}$ ). (a) Density profile with LED-CUSP scheme at  $t = 0.2, 0.4, 0.6, 0.8, 1.0$ . (b) Density profile with LED-CUSP scheme. (c) Density profile with LED-BGK scheme.



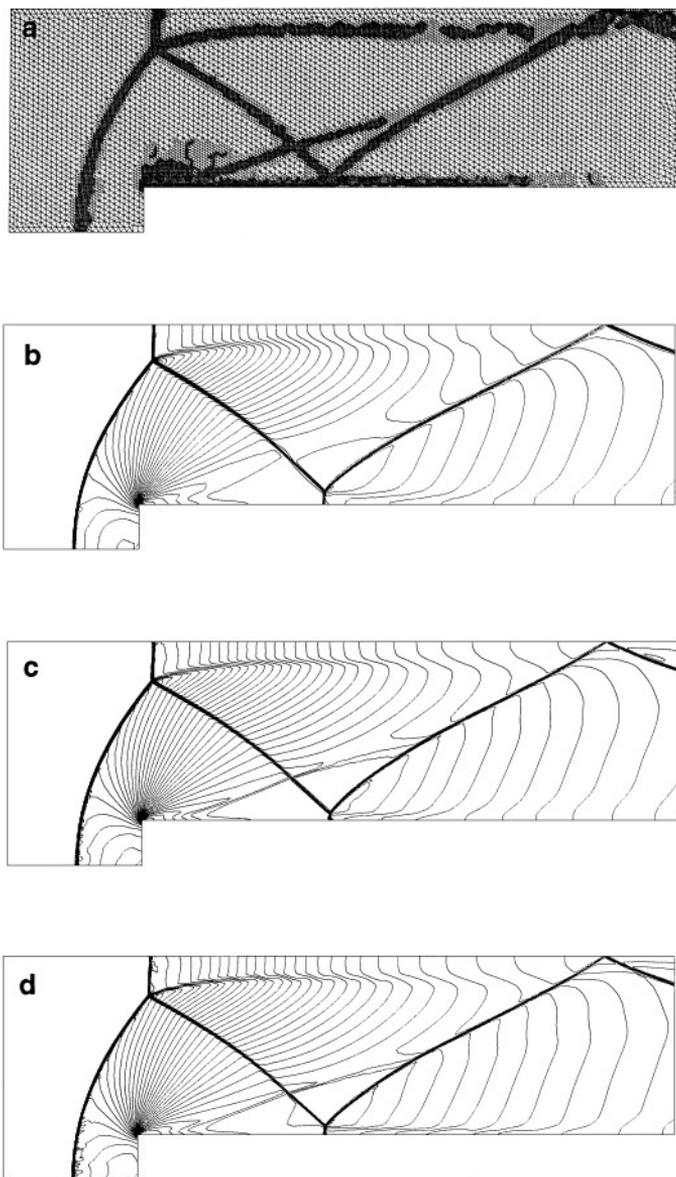
**FIG. 5.** Adapted mesh distribution. (a)  $t = 0.2$ ; 3503 nodes. (b)  $t = 0.6$ ; 3474 nodes. (c)  $t = 1.0$ ; 3534 nodes.



**FIG. 6.** Typical mesh distribution.

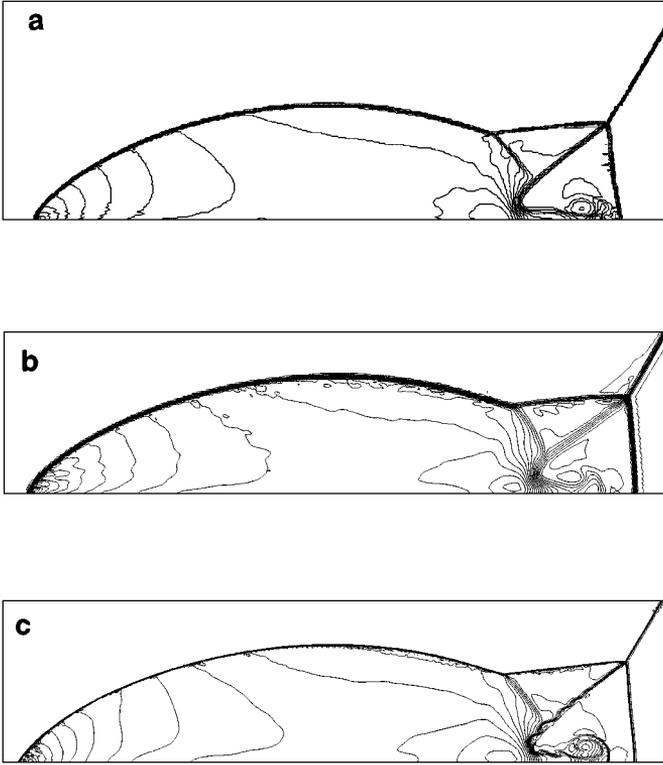


**FIG. 7.** Density distribution with LED-BGK scheme at  $t = 4.0$ . (a) Structured meshes (16,140 cells). (b) Unstructured meshes (18,110 nodes).



**FIG. 8.** Forward facing step. LED schemes at  $t = 3.0$ . (a) Typical adapted mesh distribution. (b)–(d) Density contours with (b) LED-Csplite scheme, (c) LED-CUSP scheme, and (d) LED-BGK scheme.

of the contact discontinuity and the wall jet merits particular attention. Figures 10–12 show local views of density and entropy contours, and Fig. 17 shows the mesh distribution around the triple point. Due to the increased resolution, the local flow pattern can be seen in detail. Since the contact discontinuity is unstable, the wavy motion originated from the triple point is growing, leading to a non-similar Kelvin–Helmholtz rollup along the principle slip line. The normal shock reflected from the solid wall is slightly curved due to the shock–vortex interaction. The LED-Csplite and LED-BGK schemes yield extremely good results, while the LED-CUSP scheme is slightly diffusive. Because of the orientation of the local mesh, the unsteady carbuncle phenomenon of the LED-Csplite scheme which can be observed on

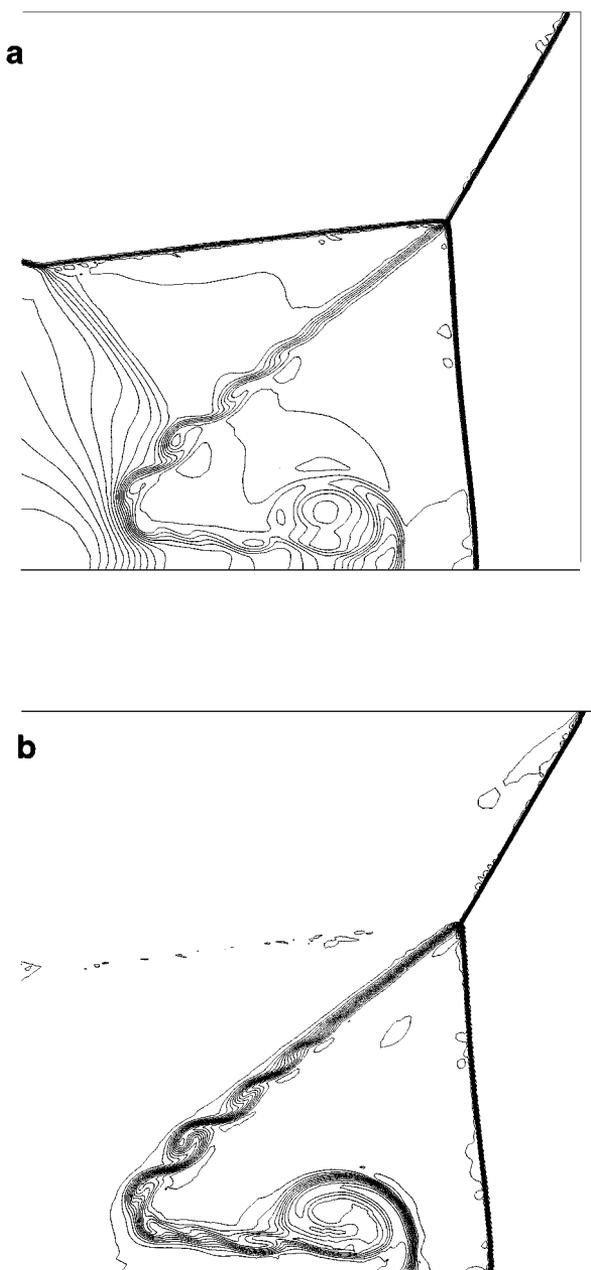


**FIG. 9.** Double Mach reflection. LED-BGK scheme. (a) Structured meshes ( $360 \times 120$  cells). (b) Unstructured meshes (17,000 nodes). (c) Unstructured adapted meshes (36,245 nodes).

a structured mesh is eliminated [18]. Similar numerical results on the instability of the contact discontinuity have been reported by Berger and Colella. These results are based on a PPM solver combined with AMR (automatic mesh refinement) technique on a rectangular mesh [6].

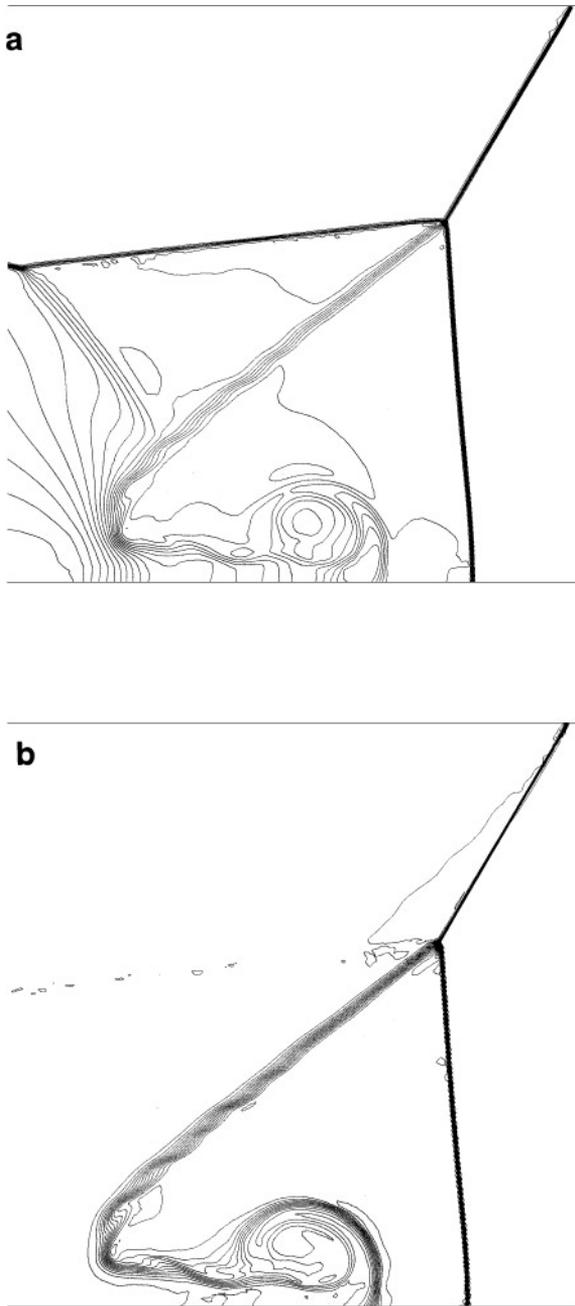
In the fourth test case, a moving shock interaction over a circular cylinder is examined. A moving shock with the speed  $M_s = 2.81$  passes the circular cylinder, which induces shock–shock and shock–contact discontinuity interactions, as well as high expansions terminated by weak shocks. Figures 13 and 14 show the density and mesh distribution at  $t = 0.55$ . The adapted mesh is slightly asymmetric with respect to the center line because of the asymmetry of the initial mesh. The LED-BGK produces equally good results compared to those of the LED-Csplit and LED-CUSP schemes.

Finally, we consider the case of a hypersonic inviscid flow around the conceptual geometry of high speed engine intake with free stream Mach number 8. The angle of the first and second cones is  $25^\circ$  and  $35^\circ$ , respectively. The initial mesh contains 2356 nodes, and the final adapted mesh contains 12,156 nodes for the LED-CUSP scheme and 12,161 nodes for the LED-BGK scheme. Figure 15 shows density contours with 40 levels, and Fig. 18 shows those for the mesh distribution at  $t = 0.4$ . The interaction of two shocks originating from the nose of each cone produces a strong contact discontinuity. The density expansion fan at the exit of the second cone is partially kinked due to the interference of the contact discontinuity. All these features are well resolved by the LED-BGK and LED-CUSP schemes. Figure 16 shows schlieren images, taken at the Princeton Gas Dynamics Laboratory [26], of the flow pattern



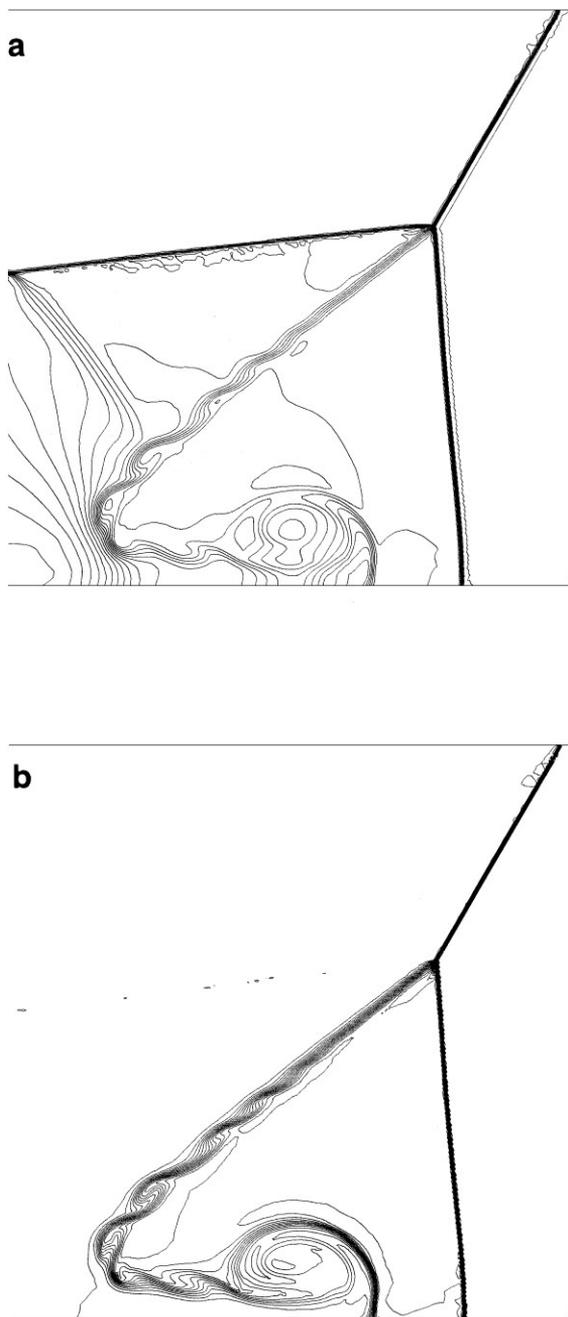
**FIG. 10.** Local view of double Mach reflection. LED-Csplit scheme. (a) Density contours. (b) Entropy contours.

around the same double cone. The images clearly show the flow characteristics that were captured computationally. However, first order characteristic splitting, as well as the higher order LED-Csplit scheme, fails to capture this flow pattern due to the problem of negative pressure and/or density. This failure can be observed regardless of mesh adaptation. The robustness of characteristic splitting is further degraded without an entropy fix. The instability of propagating normal shocks is more pronounced and physical variables in expansion regions become easily negative unless a proper amount of numerical dissipation is introduced.



**FIG. 11.** Local view of double Mach reflection. LED-CUSP scheme. (a) Density contours. (b) Entropy contours.

Although the present work is mainly directed toward the design of robust and accurate numerical methods, computational efficiency is also important. This issue may be critical in the computation of large scale three-dimensional flows. Thus it is relevant to compare the CPU time required to calculate the flux functions including time evolution terms. If the CPU time of the CUSP splitting is normalized as a unit, the ratio of CPU time required per

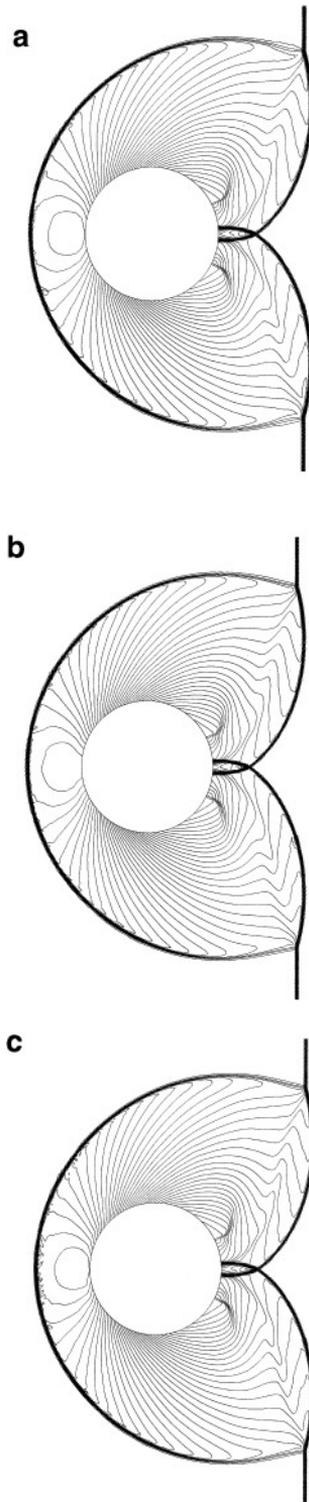


**FIG. 12.** Local view of double Mach reflection. LED-BGK scheme. (a) Density contours. (b) Entropy contours.

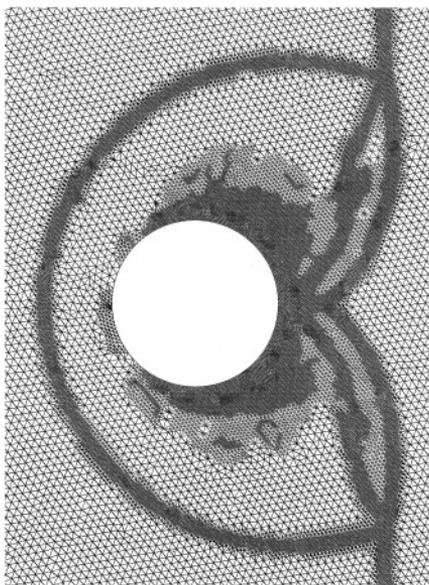
mesh point per iteration is given by

$$\text{CUSP} : \text{Csplit} : \text{BGK-I} : \text{BGK-II} = 1 : 1.136 : 1.684 : 1.908.$$

BGK-II stands for the full BGK scheme which takes account of both spatial slopes and



**FIG. 13.** Moving shock over a cylinder. Density contours with (a) LED-Csplit scheme. (b) LED-CUSP scheme. (c) LED-BGK scheme.

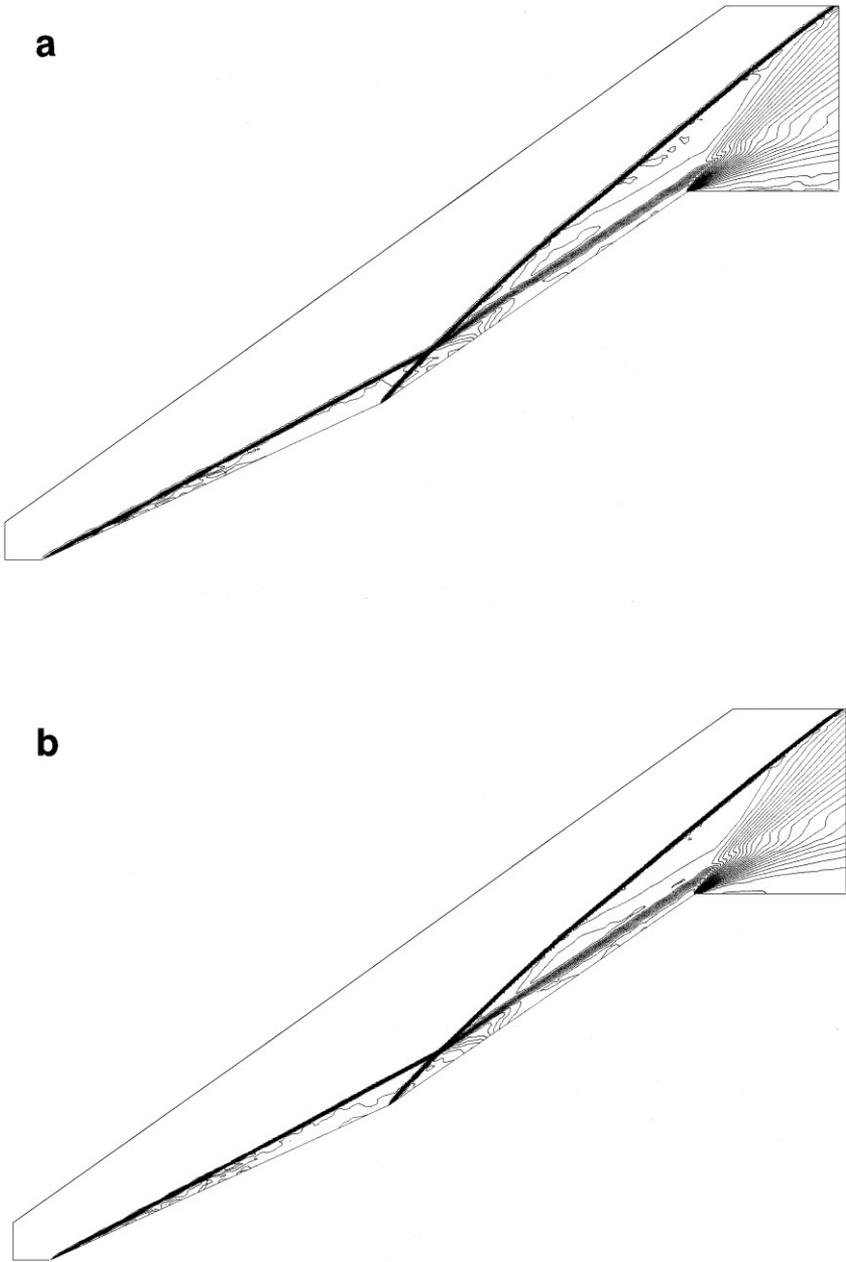


**FIG. 14.** Moving shock over a cylinder. Typical adapted mesh distribution.

BGK-I stands for the simplified BGK scheme which considers the temporal slope only. As can be seen from the ratio, the CUSP splitting is quite efficient, while the BGK scheme is relatively expensive. The reason is that unlike the CUSP or characteristic splitting, spatial slopes have to be used in the BGK solver, which inevitably needs extra arithmetical operations. Since the spatial slopes of the BGK scheme correspond to Navier–Stokes terms, the BGK solver does not need an extra flux routine for Navier–Stokes calculations. This will reduce the computational ratio substantially in Navier–Stokes computations. This issue will be addressed in future work. In addition, as can be judged from previous test cases, the superior robustness and accuracy of the LED-BGK scheme may offset the burden of extra computational costs.

## 5. CONCLUSIONS

Within a finite volume framework, a robust and accurate LED-BGK solver has been developed for unstructured adaptive meshes. In the initial reconstruction step, the LED interpolation has been employed to obtain left and right values across the cell interface. In the gas evolution step, the integral solution of the collisional BGK model is used to compute the numerical fluxes. It has been observed that the BGK model provides a good alternative to Riemann solvers for the gas evolution step. Numerical results indicate that the LED-BGK solver captures both physical discontinuities and expansion regions very well without spurious oscillations. Numerical comparison with the Roe-averaged characteristic splitting and Jameson's CUSP splitting also confirms the accuracy and robustness of the current BGK solver. The  $h$ -refinement mesh adaptation method greatly improves the resolution of physical discontinuities and shows the details of local flow patterns. This validates the capability of the current numerical approach to provide a highly accurate solution in complex geometry.



**FIG. 15.** Hypersonic double cone flow at  $t=0.4$ . Density contours with (a) LED-CUSP scheme and (b) LED-BGK scheme.

## APPENDIX

### Moments and Recurrence Relations of Distribution Functions

The formula for the evaluation of the Maxwellian distribution with the limits of  $(0, \pm\infty)$  or  $(-\infty, +\infty)$  are presented. The Maxwellian distribution of  $D$ -dimensional flow is

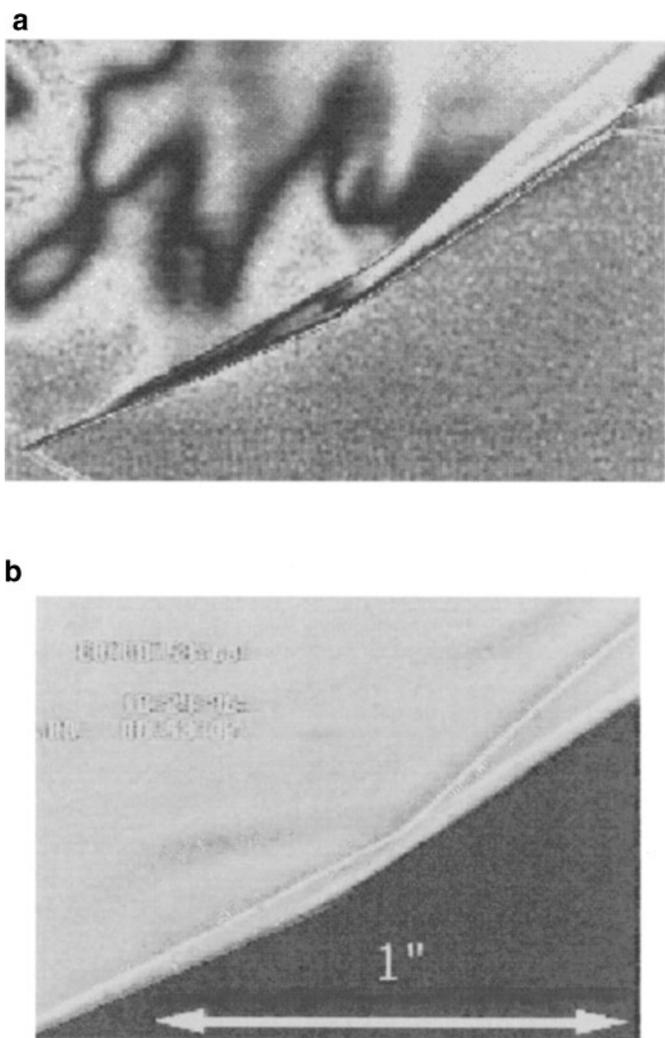


FIG. 16. Schlieren photographs of hypersonic double cone flow. (a) Wide view. (b) Local view.

given by

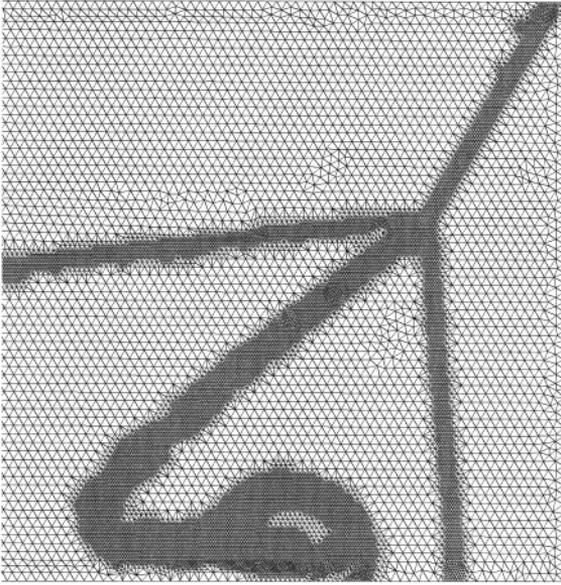
$$g = \rho \left( \frac{\lambda}{\pi} \right)^{(K+D)/2} e^{-\lambda[(u_i - U_i)^2 + \xi^2]},$$

where  $\xi$  has  $K$  degrees of freedom and  $i$  is the summation variable of  $1 \leq i \leq D$ . After introducing the definition for the moments of  $g$  with respect to  $q$  as

$$\langle q, g \rangle \equiv \int q \frac{g}{\rho} d\Xi,$$

with  $d\Xi = du_1 \cdots du_D d\xi$ , the moments of  $\xi$  appearing in the evaluation of distribution functions are given by

$$\langle \xi^2, g \rangle = \frac{K}{2\lambda}, \quad \langle \xi^4, g \rangle = \frac{K(K+2)}{4\lambda^2}, \quad \langle \xi^6, g \rangle = \frac{K(K+2)(K+4)}{8\lambda^3}.$$



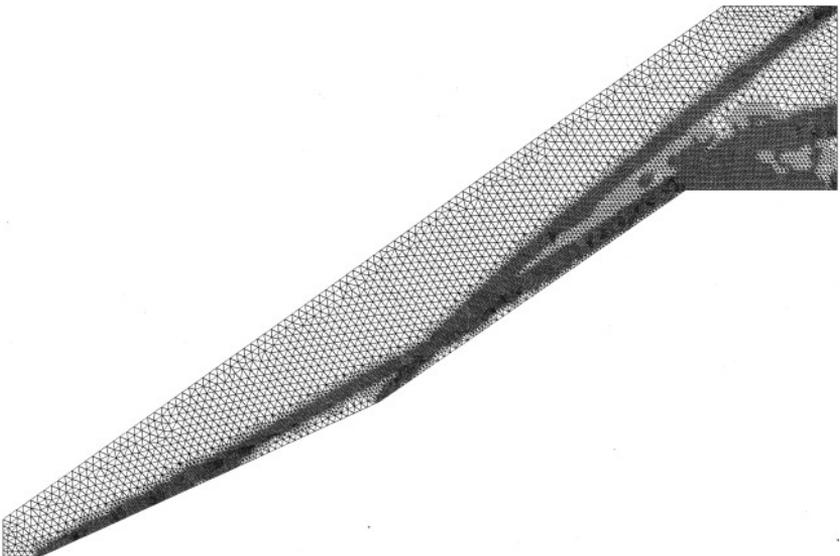
**FIG. 17.** Double Mach reflection. Typical adapted mesh distribution.

Also,

$$\begin{aligned}\langle u_j^0, g \rangle &= 1, \\ \langle u_j, g \rangle &= U_j,\end{aligned}$$

and

$$\langle u_j^{n+2}, g \rangle = U_j \langle u_j^{n+1}, g \rangle + \frac{n+1}{2\lambda} \langle u_j^n, g \rangle, \quad (25)$$



**FIG. 18.** Hypersonic double cone flow. Typical adapted mesh distribution.

if  $n \geq 2$ . The recurrence relation is simply the result of integration by parts of  $\langle u_j^n, g \rangle$  with respect to  $u_j$ . When the integral is bounded as  $(0, \pm\infty)$ , the error function or the complementary error function appears in the formula. Similarly defining the moments of  $g$  as

$$\langle q, g \rangle^+ = \int_{u_j=0}^{+\infty} q \frac{g}{\rho} d\Xi$$

or

$$\langle q, g \rangle^- \equiv \int_{u_j=-\infty}^0 q \frac{g}{\rho} d\Xi,$$

we get

$$\begin{aligned} \langle u_j^0, g \rangle^\pm &= \frac{1}{2} \operatorname{erfc}(\pm\sqrt{\lambda}U_j), \\ \langle u_j, g \rangle^\pm &= U_j \langle u_j^0, g \rangle^\pm \pm \frac{e^{-\lambda U_j^2}}{2\sqrt{\pi\lambda}}, \end{aligned}$$

and

$$\langle u_j^{n+2}, g \rangle^\pm = U_j \langle u_j^{n+1}, g \rangle^\pm + \frac{n+1}{2\lambda} \langle u_j^n, g \rangle^\pm. \quad (26)$$

With the aid of Eqs. (25) and (26), general moments of distribution functions can be calculated as

$$\langle u_j^n u_k^m \xi^l, g \rangle = \langle u_j^n, g \rangle \langle u_k^m, g \rangle \langle \xi^l, g \rangle$$

and

$$\langle u_j^n u_k^m \xi^l, g \rangle^\pm = \langle u_j^n, g \rangle^\pm \langle u_k^m, g \rangle^\pm \langle \xi^l, g \rangle^\pm.$$

In deriving the above properties, the following relations are useful:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-u^2} du, \quad \operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-u^2} du,$$

and

$$\begin{aligned} \operatorname{erf}(-x) &= -\operatorname{erf}(x), & \operatorname{erf}(0) &= 0, & \operatorname{erf}(+\infty) &= 1, \\ \operatorname{erfc}(0) &= 0, & \operatorname{erfc}(+\infty) &= 0. \end{aligned}$$

#### ACKNOWLEDGMENTS

The first author thanks Professor Luigi Martinelli of Princeton University for his valuable advices in this work. We thank Dr. T. J. Baker for useful comments on mesh generation and adaptation. We also thank referees for reviewing an earlier version of this paper carefully. This research was supported by Grant URI/AFOSR F49620-93-1-0427.

## REFERENCES

1. P. Arminjon and A. Dervieux, INRIA Report 1111, 1989 (unpublished).
2. T. J. Baker, Triangulations, mesh generation and point placement strategies, in *Proceedings Frontiers of Computational Fluid Dynamics*, edited by Caughey (1994), p. 101.
3. T. J. Baker, *Finite Elem. Anal. Des.* **25**, 243 (1997).
4. P. P. Baum and R. Löhner, in *AIAA 22nd Fluid Dynamics Conference*, 1991, 91-1666 (unpublished).
5. M. J. Berger and A. Jameson, *AIAA J.* **82**, 561 (1985).
6. M. J. Berger and P. Colella, *J. Comput. Phys.* **82**, 64 (1989).
7. M. Delanyne and J. A. Essers, in *12th AIAA CFD Conference*, 1995, CP-95-1710 (unpublished).
8. S. M. Deshpande, NASA Langley Tech. Paper No. 2613 (1986).
9. B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögren, *J. Comput. Phys.* **92**, 273 (1991).
10. S. K. Godunov, *Math. Sbornik* **47**, 271 (1959).
11. A. Harten, *J. Comput. Phys.* **49**, 357 (1983).
12. A. Harten, B. Enquist, S. Osher, and S. Chakravarthy, *J. Comput. Phys.* **171**, 231 (1987).
13. A. Harten, P. D. Lax, and B. Van Leer, *SIAM Rev.* **25**, 35 (1983).
14. A. Jameson, T. J. Baker, and N. P. Weatherhill, AIAA paper 85-0121, 1985 (unpublished).
15. A. Jameson, The present status, challenges, and future developments in computational fluid dynamics, in *AGARD 77th Fluid Dynamics Panel Symposium* (1995).
16. A. Jameson, *Int. J. Num. Methods Fluids* **20**, 743 (1996).
17. C. Johnson and P. Hansboro, *Comput. Methods Appl. Mech. Eng.* **101**, 143 (1992).
18. C. A. Kim and A. Jameson, in *12th AIAA CFD Conference*, 1995, CP-95-1738 (unpublished).
19. C. A. Kim, K. Xu, L. Martinelli, and A. Jameson, *Int. J. Num. Methods Fluids* **25**, 21 (1997).
20. C. A. Kim, A. Jameson, L. Martinelli, and K. Xu, AIAA paper 97-0328, 1997 (unpublished).
21. C. A. Kim, Ph.D. thesis, Princeton Univ. Princeton, 1997 (unpublished).
22. M. N. Kogan, *Rarefied Gas Dynamics* (Plenum Press, New York).
23. R. Löhner, *Comput. Methods Appl. Mech. Eng.* **61**, 323 (1987).
24. R. Löhner, K. Morgan, J. Peraire, and M. Vahdati, *Int. J. Num. Methods Fluids* **7**, 1093 (1987).
25. M. N. Macrossan and R. I. Oliver, *Int. J. Num. Methods Fluids* **17**, 177 (1993).
26. T. Magruder and A. J. Smits, private communication (1997).
27. J. C. Mandal and S. M. Deshpande, *Comput. Fluids* **23**, 447 (1994).
28. T. J. Mitty, T. J. Baker, and A. Jameson, *Comput. Fluids* **22**, 271 (1993).
29. J. M. Moschetta and D. I. Pullin, *J. Comput. Phys.* **133**, 193 (1997).
30. B. Perthame, *SIAM J. Num. Anal.* **29** (1992).
31. D. I. Pullin, *J. Comput. Phys.* **34**, 231 (1980).
32. J. Quirk, *Int. J. Num. Methods Fluids* **18**, 555 (1994).
33. R. D. Rausch, J. T. Batina, and H. T. Y. Yang, in *31st Aerospace Sciences Meeting and Exhibit*, 1993, 93-0670 (unpublished).
34. S. Rebay, *J. Comput. Phys.* **106**, 124 (1993).
35. P. L. Roe, *J. Comput. Phys.* **43**, 357 (1981).
36. C. W. Shu and S. Osher, *J. Comput. Phys.* **82**, 32 (1989).
37. S. Tatsumi, L. Martinelli, and A. Jameson, *AIAA J.* **33**, 252 (1995).
38. N. P. Weatherhill, O. Hassan, M. J. Marchant, and D. L. Marcum, in *11th AIAA CFD Conference*, 1993, CP-93-3390 (unpublished).
39. P. Woodward and P. Colella, *J. Comput. Phys.* **54**, 115 (1984).
40. K. Xu and K. H. Prendergast, *J. Comput. Phys.* **109**, 53 (1993).
41. K. Xu and K. H. Prendergast, *J. Comput. Phys.* **114**, 9 (1994).
42. K. Xu, L. Martinelli, and A. Jameson, *J. Comput. Phys.* **120**, 48 (1995).
43. K. Xu, C. A. Kim, L. Martinelli, and A. Jameson, *Int. J. Comput. Fluid Dynamics* **7**, 213 (1996).