COMM. MATH. SCI.
Vol. 1, No. 1, pp. 152–179

ADAPTIVE MESH REDISTRIBUTION METHOD BASED ON GODUNOV'S SCHEME*

BORIS N. AZARENOK[†], SERGEY A. IVANENKO[‡], AND TAO TANG[§]

Abstract. In this work, a detailed description for an efficient adaptive mesh redistribution algorithm based on the Godunov's scheme is presented. After each mesh iteration a second-order finite-volume flow solver is used to update the flow parameters at the new time level directly *without* using interpolation. Numerical experiments are performed to demonstrate the efficiency and robustness of the proposed adaptive mesh algorithm in one and two dimensions.

Key words. Adaptive mesh redistribution algorithm, moving mesh method, Godunov's scheme, hyperbolic conservation laws.

AMS subject classifications. Subject Classification: 65M93, 35L64, 76N10.

1. Introduction

Several moving mesh techniques have been introduced in the past for solving the problems governed by the hyperbolic conservation laws. Harten and Hyman [15] began the earliest study in this direction by moving the grid at an adaptive speed in each time step to improve the resolution of shocks and contact discontinuities. After their work, many other moving mesh methods for hyperbolic problems have been proposed in the literature, including Azarenok and Ivanenko [1, 19], Liu et al. [24], Mackenzie et al. [29], and Tang [36]. However, it is noticed that some existing moving mesh methods for hyperbolic problems are designed for one space dimension. In 1D, it is generally possible to compute on a very fine grid and so the need for moving mesh methods may not be clear. Multidimensional moving mesh methods are often difficult to use in fluid dynamics problems since the grid will typically suffer large distortions and possible tangling. It is therefore useful to design robust moving mesh algorithms for hyperbolic problems in multidimensions.

Several adaptive grid methods have been proposed based on the harmonic mapping theory, see e.g., [33, 34]. Dvinsky [12] suggests the possibility that harmonic function theory may provide a general framework for developing useful mesh generators. Meshes obtained by Dvinsky's method have desirable properties, particularly regularity or smoothness. Motivated by the work of Dvinsky, a moving mesh finite element strategy based on harmonic mapping was proposed and studied by Li et al. in [27, 28]. The numerical experiments show that their methods can accurately resolve detail features of singular problems in two- and three-space dimensions. Note that adapted meshes (and nonadapted meshes as well) in the complex physical domains, e.g. nonconvex, obtained by solving the discretized Euler equations, cannot be guaranteed of folding. Another class of adaptive grid generation method based on the

---

*Received: May 9, 2002; Accepted (in revised version): June 3, 2002.

[†]Computing Center of Russian Academy of Sciences, Vavilov str. 40, GSP-1, Moscow, Russia (azarenok@ccas.ru).

[‡]Computing Center of Russian Academy of Sciences, Vavilov str. 40, GSP-1, Moscow, Russia (sivan@ccas.ru).

[§]Department of Mathematics, The Hong Kong Baptist University, Kowloon Tong, Hong Kong and Institute of Computational Mathematics, Chinese Academy of Sciences, Beijing 100080, China (ttang@math.hkbu.edu.hk, ttang@lsec.cc.ac.cn).

harmonic maps has been suggested by Ivanenko et al. [17, 8, 18]. Their method is to minimize a finite-difference approximation to a Dirichlet's (or harmonic) functional. The discrete functional has an infinite barrier at the boundary of the set of grids with all convex quadrilateral cells. The infinite barrier guarantees unfolded grid generation in any simply and multiply connected 2D domains. The barrier property is also of particular importance in the vicinity of shock waves where the cells may become very narrow [3].

In this work, we will describe an adaptive mesh redistribution (AMR) method based on minimizing a harmonic functional. The AMR method combines the use of the Godunov-type solver and adaptive moving mesh. Some ideas were implemented in [1, 19]. The main strategies of this approach are twofold. First, after each mesh iteration, a second-order finite-volume flow solver, suggested by Azarenok [2], updates the flow parameters at the new time level directly on the adaptive grid without using interpolation. This is in contrast with the work of Tang et al. [36, 27, 28]. Second, we will solve some *optimization problem* in order to obtain the mesh, which is in contrast to the traditional method of solving the Euler-Lagrange equations directly [5, 6, 27]. Using an optimization approach to obtain the mesh has been also realized recently by Li et al. [28] in their moving mesh finite element methods. The key principle of the present approach is to minimize some discrete Dirichlet's functional possessing the infinite barrier property, as suggested by Charakhch'yan and Ivanenko [7]. It is noted that the recent paper of Knupp et al. [20] used the same idea of [7].

An outline of the paper is as follows. In Section 2, we describe the flow solver to calculate ideal gas flow on a moving mesh. In Section 3, we describe the problem formulation of grid generation and in Section 4, the numerical algorithm for constructing an adaptive mesh. In Section 5, numerical examples will be presented.

2. Flow Solver

In this section we describe the Godunov Linear Flux Correction (GLFC) scheme for computing the 1D and 2D ideal gas flow on a moving mesh [2]. This scheme on one hand utilizes the idea of the Godunov's scheme on the deforming meshes [14] and on the other hand is of second-order accuracy in both time and space. We begin with 2D case.

2.1. System of Equations.

The governing system of the differential equations relating to 2D gas flow is

$$\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial \boldsymbol{a}}{\partial x} + \frac{\partial \boldsymbol{b}}{\partial y} = \boldsymbol{c}\,,$$

where the vector-valued functions are defined by

$$\begin{aligned}&\boldsymbol{\sigma}=(\rho,\rho u,\rho v,E)^\top, \quad \boldsymbol{a}=(\rho u,\rho u^2+p,\rho uv,u(E+p))^\top,\\ &\boldsymbol{b}=(\rho v,\rho uv,\rho v^2+p,v(E+p))^\top, \quad \boldsymbol{c}=-\frac{\nu v}{y}(\rho,\rho u,\rho v,E+p)^\top.\end{aligned} \tag{2.1}$$

u and v are the velocity components, p and ρ are the pressure and density, respectively, $\nu=0,1$ represent the planar and axi-symmetric flow, respectively. The total energy $E=\rho[e+0.5(u^2+v^2)]$, e is the specific internal energy. The equation of state is $p=(\gamma-1)\rho e$, where γ is the ratio of the specific heats. We denote the vector-valued function of flow parameters as $\boldsymbol{f}=(u,v,p,\rho)^T$.

We perform calculations utilizing the integral conservation laws which can be derived by integrating the above system of the differential equations and transforming the volume integrals in space $x{-}y{-}t$ to the surface integrals by virtue of Gauss's theorem

$$\iiint_V \left[\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial \boldsymbol{a}}{\partial x} + \frac{\partial \boldsymbol{b}}{\partial y} - \boldsymbol{c}\right] dV = \oiint_{\partial V} \boldsymbol{\sigma} dxdy + \boldsymbol{a} dydt + \boldsymbol{b} dtdx - \iiint_V \boldsymbol{c} dxdydt = 0.$$

Here, V is an arbitrary control volume, homeomorphic sphere in space $x{-}y{-}t$, ∂V is the boundary of V. Hence, the laws of conservation of mass, momentum, and total energy can be written in the integral form, or generalized formulation, as follows [14]

$$\oiint_{\partial V} \boldsymbol{\sigma} dxdy + \boldsymbol{a} dydt + \boldsymbol{b} dtdx = \iiint_V \boldsymbol{c} dxdydt, \tag{2.2}$$

The conservation laws (2.2) hold for any parameters $\boldsymbol{f}$, both smooth and discontinuous and therefore govern a real gas flow.

2.2. Discretization of Equations. We introduce the curvilinear moving grid in $\mathcal{R}^3$ space x-y-t, a hexahedral cell of which with number $(i+2/2, j+1/2)$ is depicted in Fig. 2.1. The bottom face of the cell (or control volume) is taken at time level n and the top face at level $n+1$, while four lateral faces generally form ruled surfaces rather than simple planes.

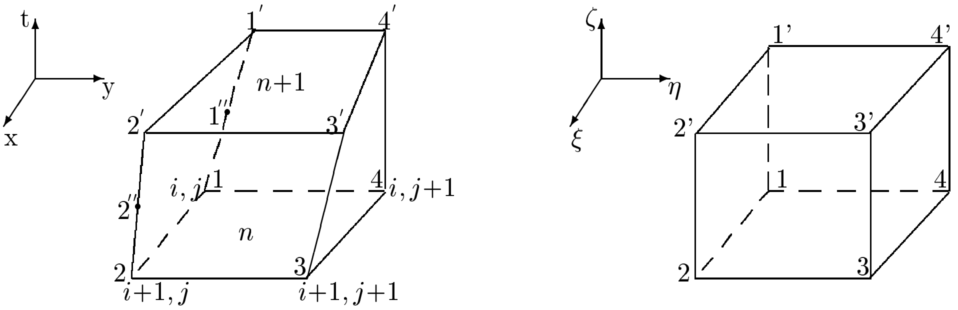

Fig. 2.1. *Hexahedral cell in the physical space (left) and the parametric space (right).*

Integrating (2.2) over the oriented surface, i.e., the boundary ∂V of the computing cell, gives a cell-centered finite-volume discretization of the governing equations

$$\boldsymbol{\sigma}^{n+1} A^{n+1} - \boldsymbol{\sigma}^n A^n + \boldsymbol{Q}_{411'4'} + \boldsymbol{Q}_{233'2'} + \boldsymbol{Q}_{122'1'} + \boldsymbol{Q}_{433'4'} = \boldsymbol{c}^{n+1/2} V, \tag{2.3}$$

where $\boldsymbol{\sigma}^{n+1}$ and $\boldsymbol{\sigma}^n$ are the average values at time t^{n+1} and t^n in the center of the top and bottom faces, respectively; A^{n+1} and A^n are the areas of the corresponding faces. Each of the four vector values $\boldsymbol{Q}_{411'4'}$, $\boldsymbol{Q}_{233'2'}$, $\boldsymbol{Q}_{122'1'}$ and $\boldsymbol{Q}_{344'3'}$ is an average flux of mass, momentum, and energy through the corresponding intercell surface in the direction of the outward normal vector. Unlike the original Godunov's scheme [14] where the fluxes in (2.3) are taken at time t^n, in the present scheme those values are computed at $t^{n+1/2}$, which provides the second-order accuracy in time.

For example, at the face $122'1'$, the value of $\boldsymbol{Q}_{122'1'}$ has the following structure:

$$\boldsymbol{Q}_{122'1'} = \boldsymbol{\sigma}^{n+1/2} A^{xy} + \boldsymbol{a}^{n+1/2} A^{yt} + \boldsymbol{b}^{n+1/2} A^{tx}, \tag{2.4}$$

where $\boldsymbol{\sigma}^{n+1/2}, \boldsymbol{a}^{n+1/2}, \boldsymbol{b}^{n+1/2}$ are calculated using parameters $\boldsymbol{f}^{n+1/2}$ in the center of the face, i.e., at the mid-point of edge 12 at time $t^{n+1/2}$; A^{xy}, A^{yt}, A^{tx} are the areas of projections of the face $122'1'$ onto the coordinate planes x-y, y-t, and t-x, respectively, given by

$$\begin{aligned} A^{xy} &= \iint_{122'1'} dxdy = 0.5[(x_{2'} - x_1)(y_{1'} - y_2) - (x_{1'} - x_2)(y_{2'} - y_1)], \\ A^{yt} &= \iint_{122'1'} dydt = 0.5\Delta t(y_{2'} + y_2 - y_1 - y_{1'}), \\ A^{tx} &= \iint_{122'1'} dtdx = -0.5\Delta t(x_{2'} + x_2 - x_1 - x_{1'}), \end{aligned} \tag{2.5}$$

where the time step $\Delta t = t^{n+1} - t^n$. The equations (2.5) are obtained from the formula for the quadrangle 1234

$$A_{1234} = A(x_1, y_1; x_2, y_2; x_3, y_3; x_4, y_4) = 0.5[(x_3 - x_1)(y_4 - y_2) - (x_4 - x_2)(y_3 - y_1)]$$

when running along its contour in an anticlock wise manner.

The vector-valued function $\boldsymbol{c}^{n+1/2}$ is defined in the center of the quadrilateral cell at time $t^{n+1/2}$ (i.e., in the center of the control volume), V is the volume of the hexahedral cell. Formulas for the control volume V of a ruled/(hexahedral) cell were derived in [11, 38]. For instance, V is computed by integrating over the control volume surface ∂V by Dukowicz [11]. The contribution to the total volume V from the surface 1234, referred to as V_{1234}, obtained by integrating over the bottom face 1234 is

$$\begin{aligned} V_{1234} &= [\boldsymbol{r}_1 \cdot (\boldsymbol{r}_4 \times \boldsymbol{r}_3) + \boldsymbol{r}_2 \cdot (\boldsymbol{r}_1 \times \boldsymbol{r}_4) + \boldsymbol{r}_3 \cdot (\boldsymbol{r}_2 \times \boldsymbol{r}_1) + \boldsymbol{r}_4 \cdot (\boldsymbol{r}_3 \times \boldsymbol{r}_2)]/12 \\ &= (\boldsymbol{r}_1 + \boldsymbol{r}_2) \cdot [(\boldsymbol{r}_1 + \boldsymbol{r}_4) \times (\boldsymbol{r}_2 + \boldsymbol{r}_3)]/12, \end{aligned} \tag{2.6}$$

where $\boldsymbol{r}_i = (x, y, t)_i$. Formulas for the contribution of the other five cell faces can be obtained from (2.6) by cyclic permutation of indexes. In [38] it is proved that V equals the mean of volumes of two dodecahedrons with planar faces which have the same vertices as the hexahedral cell.

To determine $\boldsymbol{c}^{n+1/2}$ we need also the coordinate y_c of the cell center which is obtained from trilinear mapping forming the hexahedral cell, see FIG. 2.1. At $\xi, \eta, \zeta = 0.5$ it is given by

$$y_c = \sum_i y_i/8, \qquad i = 1, \cdots, 4, 1', \cdots, 4'.$$

The values $\boldsymbol{f}^{n+1}$ are updated by two stages using a time splitting technique proposed in [31], which is of second-order accuracy in time. It is pointed out that another scheme with second-order accuracy in time has been suggested in [4] where the Riemann problem is solved based on piecewise linear distribution of the flow parameters (generalized Riemann problem). At the first stage, predictor, via (2.3) we compute the intermediate values at the n+1-th level $\bar{\boldsymbol{f}}^{n+1}$. Here we apply the piecewise linear interpolation along each curvilinear coordinate line ξ, passing through the center of cells $j+1/2$=*const.*; and η, passing through the center of cells $i+1/2$=*const.*, i.e., determine the derivatives $\boldsymbol{f}_\xi$ and $\boldsymbol{f}_\eta$ in every cell to get the fluxes via (2.4) on the lateral faces with the second-order accuracy in space but still at t^n. To suppress spurious oscillations in the vicinity of discontinuities, a monotonicity algorithm will be applied.

2.3. Determination of Fluxes.

For clarity let us determine the fluxes in 1D case; in 2D case such an 1-D algorithm is executed along each of the curvilinear coordinate, considered as a parameter of length. Consider a cell of a 1D grid along the curvilinear coordinate ξ as shown in FIG. 2.2. Assume the function $\boldsymbol{f}$ to be linear within the zone $(i, i{+}1)$. For simplicity we omit the superscript n: If the superscript is not specified, then all the values are taken at t^n. The values $\boldsymbol{f}_{1,i+1/2}$ and $\boldsymbol{f}_{2,i+1/2}$, specified at the left and right zone ends are defined by

$$\boldsymbol{f}_{1,i+1/2} = \boldsymbol{f}_{i+1/2} - 0.5\delta\boldsymbol{f}_{i+1/2}h_{i+1/2} \;\;, \quad \boldsymbol{f}_{2,i+1/2} = \boldsymbol{f}_{i+1/2} + 0.5\delta\boldsymbol{f}_{i+1/2}h_{i+1/2}. \tag{2.7}$$

Here, the spacing along the ξ coordinate line at fixed index j is given by

$$h_{i+1/2} = 0.5\sqrt{(x_{i+1,j} + x_{i+1,j+1} - x_{i,j} - x_{i,j+1})^2 + (y_{i+1,j} + y_{i+1,j+1} - y_{i,j} - y_{i,j+1})^2}.$$

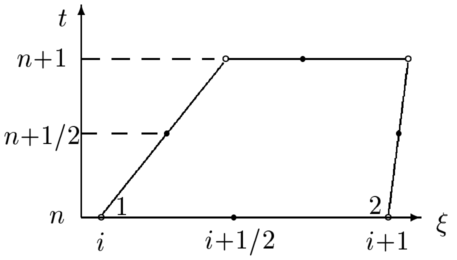

FIG. 2.2. *Computing cell of 1D grid along the curvilinear coordinate* ξ*.*

The “effective” derivative $\delta\boldsymbol{f}_{i+1/2}$ is defined via the auxiliary values $\tilde{\boldsymbol{f}}_{1,i+1/2}$ and $\tilde{\boldsymbol{f}}_{2,i+1/2}$ at the right and left zone ends

$$\delta\boldsymbol{f}_{i+1/2} = (\tilde{\boldsymbol{f}}_{2,i+1/2} - \tilde{\boldsymbol{f}}_{1,i+1/2})/h_{i+1/2}. \tag{2.8}$$

The auxiliary values at the left zone end $\tilde{\boldsymbol{f}}_{1,i+1/2}$ and the right zone end $\tilde{\boldsymbol{f}}_{2,i+1/2}$ are obtained by using the Taylor expansion about the points ξ_i and ξ_{i+1}, respectively,

$$\begin{aligned}
\boldsymbol{f}_{i+1/2} &= \boldsymbol{f}_i + \frac{1}{2}h_{i+1/2}\boldsymbol{f}^{(1)}_i + \frac{1}{8}h^2_{i+1/2}\boldsymbol{f}^{(2)}_i + \mathcal{O}(h^3_{i+1/2}),\\
\boldsymbol{f}_{i+1/2} &= \boldsymbol{f}_{i+1} - \frac{1}{2}h_{i+1/2}\boldsymbol{f}^{(1)}_{i+1} + \frac{1}{8}h^2_{i+1/2}\boldsymbol{f}^{(2)}_{i+1} - \mathcal{O}(h^3_{i+1/2}).
\end{aligned}$$

By neglecting the higher-order terms, we obtain from the above two expressions that

$$\begin{aligned}
\tilde{\boldsymbol{f}}_{1,i+1/2} &= \boldsymbol{f}_i = \boldsymbol{f}_{i+1/2} - \frac{1}{2}h_{i+1/2}\boldsymbol{f}^{(1)}_i - \frac{1}{8}h^2_{i+1/2}\boldsymbol{f}^{(2)}_i,\\
\tilde{\boldsymbol{f}}_{2,i+1/2} &= \boldsymbol{f}_{i+1} = \boldsymbol{f}_{i+1/2} + \frac{1}{2}h_{i+1/2}\boldsymbol{f}^{(1)}_{i+1} - \frac{1}{8}h^2_{i+1/2}\boldsymbol{f}^{(2)}_{i+1}.
\end{aligned} \tag{2.9}$$

The first and second derivatives are given by the relations

$$\begin{aligned}
\boldsymbol{f}^{(1)}_i &= \frac{2}{h_{i+1/2} + h_{i-1/2}}\left(\boldsymbol{f}_{i+1/2} - \boldsymbol{f}_{i-1/2}\right),\\
\boldsymbol{f}^{(2)}_i &= \frac{2}{h_{i+1/2} + h_{i-1/2}}\left(\boldsymbol{f}^{(1)}_{i+1/2} - \boldsymbol{f}^{(2)}_{i-1/2}\right).
\end{aligned} \tag{2.10}$$

Substituting them into (2.9) gives

$$\begin{aligned}\tilde{\boldsymbol{f}}_{1,i+1/2} &= \boldsymbol{f}_{i+1/2} - \frac{h_{i+1/2}}{h_{i+1/2}+h_{i-1/2}}\left(\boldsymbol{f}_{i+1/2}-\boldsymbol{f}_{i-1/2}\right)\\ &\quad -\frac{h^2_{i+1/2}}{4(h_{i+1/2}+h_{i-1/2})}\left(\boldsymbol{f}^{(1)}_{i+1/2}-\boldsymbol{f}^{(1)}_{i-1/2}\right),\\ \tilde{\boldsymbol{f}}_{2,i+1/2} &= \boldsymbol{f}_{i+1/2} + \frac{h_{i+1/2}}{h_{i+3/2}+h_{i+1/2}}\left(\boldsymbol{f}_{i+3/2}-\boldsymbol{f}_{i+1/2}\right)\\ &\quad -\frac{h^2_{i+1/2}}{4(h_{i+3/2}+h_{i+1/2})}\left(\boldsymbol{f}^{(1)}_{i+3/2}-\boldsymbol{f}^{(1)}_{i+1/2}\right).\end{aligned} \tag{2.11}$$

2.4. Monotonicity algorithm. In (2.11) every component of the derivative $\boldsymbol{f}^{(1)}$ is defined by the Kolgan's principle of minimal derivatives [21]. For example, to the velocity u it reads

$$u^{(1)}_{i+1/2} = \begin{cases} u^{(1)}_i\,, & \text{if } |u^{(1)}_i| \le |u^{(1)}_{i+1}| \\ u^{(1)}_{i+1}\,, & \text{otherwise,}\end{cases}$$

where the derivative is given by the first relation of (2.10). Afterwards we apply the Collela-Woodward's monotonicity algorithm [9] to reset the auxiliary values $\tilde{\boldsymbol{f}}_{1,i+1/2}$ and $\tilde{\boldsymbol{f}}_{2,i+1/2}$. We construct an interpolating parabola within the zone for every component of the vector $\boldsymbol{f}$. For example, to u we have

$$U(\zeta) = a_0 + a_1\zeta + a_2\zeta^2\ ,\ \ \zeta = (\xi - \xi_{i+1/2})/h_{i+1/2}\ ,\ \ \xi_i \le \xi \le \xi_{i+1}\ ,\ \ -0.5 \le \zeta \le 0.5.$$

The parabola satisfies the following requirements:

$$U(-0.5){=}u_{1,i+1/2},\quad U(0.5){=}u_{2,i+1/2},\quad u_{i+1/2} = \int_{-0.5}^{0.5} U(\xi)d\xi,$$

where $u_{i+1/2}$ is the zone average. It can be shown that the coefficients a_1 and a_2 are

$$a_1{=}\tilde{u}_{2,i+1/2}{-}\tilde{u}_{1,i+1/2},\quad a_2{=}3(\tilde{u}_{1,i+1/2}{+}\tilde{u}_{2,i+1/2}){-}6u_{i+1/2}\,.$$

The coefficient a_0 is not needed in our computation.

A condition to be imposed is that the interpolating parabola shall not achieve its maximum or minimum within the zone. Otherwise the values $\tilde{u}_{1,i+1/2}$,$\tilde{u}_{2,i+1/2}$ have to be corrected in order to ensure that the parabola is monotone. There are two possible cases when it takes place:

a) $u_{i+1/2}$ does not lie inside the interval from $\tilde{u}_{1,i+1/2}$ to $\tilde{u}_{2,i+1/2}$. In such a case the interpolating function is set to be a constant; i.e,

$$\tilde{u}_{1,i+1/2} = \tilde{u}_{2,i+1/2} = u_{i+1/2}\ ,\ \text{if}\ \ (\tilde{u}_{2,i+1/2} - u_{i+1/2})(u_{i+1/2} - \tilde{u}_{1,i+1/2}) \le 0;$$

b) $u_{i+1/2}$ lies inside the above interval, but sufficiently close to one of the end points. Then we reset $\tilde{u}_{1,i+1/2}$ or $\tilde{u}_{2,i+1/2}$ (which is farther from $u_{i+1/2}$) so that the interpolating parabola becomes monotone as follows:

$$\begin{cases} \tilde{u}_{1,i+1/2} = u_{i+1/2} + 2(u_{i+1/2} - \tilde{u}_{2,i+1/2})\ , & \text{if}\ \ a_1^2 < -a_1a_2, \\ \tilde{u}_{2,i+1/2} = u_{i+1/2} + 2(u_{i+1/2} - \tilde{u}_{1,i+1/2})\ , & \text{if}\ \ a_1^2 < a_1a_2.\end{cases}$$

Next, from (2.8) we get $\delta \boldsymbol{f}_{i+1/2}$ and from (2.7) the zone end values $\boldsymbol{f}_{1,i+1/2}$ and $\boldsymbol{f}_{2,i+1/2}$ used to calculate the fluxes $\boldsymbol{Q}_{411'4'}$ and $\boldsymbol{Q}_{233'2'}$ via formulas being similar to (2.4). In the same manner we evaluate $\boldsymbol{f}_{1,j+1/2}$ and $\boldsymbol{f}_{2,j+1/2}$ used to calculate the fluxes $\boldsymbol{Q}_{122'1'}$ and $\boldsymbol{Q}_{433'4'}$. Further, from (2.3) we obtain the intermediate value $\bar{\boldsymbol{\sigma}}^{n+1}$ at time t^{n+1}.

At the second stage, corrector, we first get the prewave values $\boldsymbol{f}$ at the center of the intercell faces at time $t^{n+1/2}$. For it we set the effective derivative at t^{n+1} equal to the one at t^n, i.e. $\delta\bar{\boldsymbol{f}}^{n+1}_{i+1/2}=\delta\boldsymbol{f}^{n}_{i+1/2}$. Then the zone end values to the intermediate function $\bar{\boldsymbol{f}}^{n+1}_{i+1/2}$ are obtained from (2.7). Taking the mean of edge values at t^n and t^{n+1} we get the prewave states at the faces 411'4' and 233'2' at time $t^{n+1/2}$ as follows:

$$\begin{aligned}\boldsymbol{f}^{n+1/2}_{1,i+1/2} &= 0.5\left[\boldsymbol{f}^{n}_{i+1/2}+\bar{\boldsymbol{f}}^{n+1}_{i+1/2}-0.5\delta\boldsymbol{f}^{n}_{i+1/2}(h^{n}_{i+1/2}+h^{n+1}_{i+1/2})\right],\\ \boldsymbol{f}^{n+1/2}_{2,i+1/2} &= 0.5\left[\boldsymbol{f}^{n}_{i+1/2}+\bar{\boldsymbol{f}}^{n+1}_{i+1/2}+0.5\delta\boldsymbol{f}^{n}_{i+1/2}(h^{n}_{i+1/2}+h^{n+1}_{i+1/2})\right].\end{aligned} \tag{2.12}$$

Similarly we calculate $\boldsymbol{f}^{n+1/2}_{1,j+1/2}, \boldsymbol{f}^{n+1/2}_{2,j+1/2}$, the prewave states at the faces 122'1' and 433'4'.

2.5. Riemann Problem on the Moving Mesh. In this subsection, we solve the Riemann problem to each of four intercell faces, taking into account the velocity of the intercell face and using the Riemann solver suggested in [14] (for description of this Riemann solver see also [22]). Numerical calculations have shown that for 2D problems with intensive shocks using this Riemann solver based on the exact solution of the nonlinear PDE's system gives loss in computing time by only ≈5% in comparison with the time when the Riemann solver based on the linearized system (cf. [13]) is employed. The idea of taking into account the velocity of the intercell face is to be demonstrated for the 1D case. Consider the intercell boundary of the cell within the time interval $(t^{n+1/2}, t^{n+1})$. Assume that after solving the Riemann problem at point $(i, n{+}1/2)$ we have the wave pattern depicted in FIG. 2.3. There are 5 cases of location of the intercell boundary $[(i,n{+}1/2)(i,n{+}1)]$ in the wave pattern depending on velocity of the i-th node w_i. As the postwave values $\boldsymbol{f}^{n+1/2}_{i}$ we use:

1. $\boldsymbol{f}^{n+1/2}_{i}=\boldsymbol{f}^{n+1/2}_{2,i-1/2}$ if $w_i < u_{sh}$, where u_{sh} is the speed of the left shock.

2. $\boldsymbol{f}^{n+1/2}_{i}=\boldsymbol{f}_2$ if $u_{sh} < w_i < u_{cont}$, where the vector $\boldsymbol{f}_2$ defines the flow parameters after the shock, u_{cont} is the speed of the contact discontinuity.

3. $\boldsymbol{f}^{n+1/2}_{i}=\boldsymbol{f}_3$ if $u_{cont} < w_i < u^{lft}_{rar}$, where the vector $\boldsymbol{f}_3$ defines parameters in the domain between the contact discontinuity and left characteristic of the rarefaction wave expanding with the speed u^{lft}_{rar}.

4. $\boldsymbol{f}^{n+1/2}_{i}=\phi(x/t)$ if $u^{lft}_{rar} < w_i < u^{rght}_{rar}$; i.e., we calculate the flow parameters in the rarefaction wave using the variable x/t. Here, u^{rght}_{rar} is the speed of the right characteristic in the rarefaction fan.

5. $\boldsymbol{f}^{n+1/2}_{i}=\boldsymbol{f}^{n+1/2}_{1,i+1/2}$ if $w_i > u^{rght}_{rar}$.

In the 2D case, we by analogy consider the Riemann problem on the moving mesh. To get the postwave states $\boldsymbol{f}^{n+1/2}$ in the center of face $122'1'$; i.e., at the mid-point of the segment $1''2''$, (see FIG. 2.1), we solve the Riemann problem with the prewave states $(r,p,\rho)^{n+1/2}$ at this point on both sides of the face (one state relates to the cell considered and the other to the cell adjacent to the face $122'1'$), and use the tangential

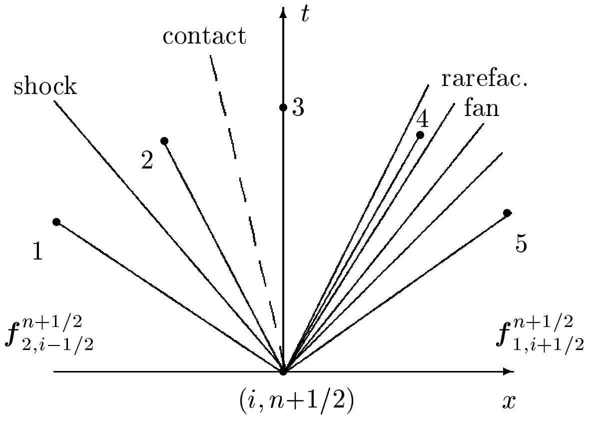

FIG. 2.3. *Five probable cases of location of the intercell boundary* $[(i,n{+}1/2)(i,n{+}1)]$ *in the wave pattern. Points* $1,\ldots,5$ *indicate the node* $(i,n{+}1)$ *location.*

components of the velocity $q^{n+1/2}$ on those sides. Here, $r^{n+1/2}$ is a normal component of the velocity. The normal and tangential components of the velocity are given by

$$r^{n+1/2} = n_x u^{n+1/2} + n_y v^{n+1/2},$$
$$q^{n+1/2} = n_y u^{n+1/2} - n_x v^{n+1/2},$$

where n_x and n_y are the components of the outward normal vector to the segment $1''2''$.

After solving the Riemann problem, we find the tangential component $q^{n+1/2}$ via

$$q^{n+1/2} = \begin{cases} q^{n+1/2} & \text{if } \; w_{12} \le u_{cont}, \\ \tilde{q}^{n+1/2} & \text{otherwise,} \end{cases} \tag{2.13}$$

where u_{cont} is the speed of the contact discontinuity in the Riemann problem, w_{12} is the velocity of the edge 12 in the normal direction to this edge,$\tilde{q}^{n+1/2}$ is the tangential component of the velocity in the cell adjacent to the face $122'1'$. The condition (2.13) expresses the fact that the tangential component of the velocity is discontinuous across the tangential (contact) discontinuity; see e.g., [10]. The velocity w_{12} can be defined from the equality

$$\Delta t l_{1''2''} w_{12} = A^{xy}, \tag{2.14}$$

where $l_{1''2''}$ is the distance between the points $1''$ and $2''$. We restore the Cartesian components of the velocity in the center of the face

$$u^{n+1/2} = n_x r^{n+1/2} + n_y q^{n+1/2},$$
$$v^{n+1/2} = n_y r^{n+1/2} - n_x q^{n+1/2}.$$

In calculating $\boldsymbol{c}^{n+1/2}$ the average velocity in the $(i{+}1/2, j{+}1/2)$ point is given by

$$u^{n+1/2} = 0.5(u^n + \bar{u}^{n+1}),$$

where $\bar{u}^{n+1}$ is the intermediate value obtained at the predictor stage. Similarly we can obtain $(v, p, \rho)^{n+1/2}$.

After solving the Riemann problem, we obtain the postwave values $\boldsymbol{f}$ at the center of 4 intercell faces. Calculating the fluxes through the intercell face 122'1' via (2.4) and similarly through the other 3 intercell faces, and substituting the values of the fluxes into (2.3), we get the final values of mass, momentum, and energy $\boldsymbol{\sigma}^{n+1}$ at time t^{n+1}.

Note that, when calculating by (2.11) the auxiliary values, at the boundary nodes we should use the boundary conditions. For instance, if at the left node, i=0, the boundary condition is defined as $\boldsymbol{f}_{left}$, then we set $\tilde{\boldsymbol{f}}_{1,1/2}$=$\boldsymbol{f}_{left}$. If the boundary condition is defined to the fluxes, e.g. reflecting boundary condition, we set $\tilde{\boldsymbol{f}}_{1,1/2}$=$\boldsymbol{f}_0^{n-1/2}$, where $\boldsymbol{f}_0^{n-1/2}$ are the postwave values at the preceding time step.

2.6. Stability Condition. In 1D case the choice of admissible time step Δt has a clear physical sense. On the moving grid Δt is given by

$$\Delta t = c_{cfl} \min_i \Delta t_{i+1/2} \tag{2.15}$$

where in every cell as shown in FIG. 2.2 the local time step is determined by (cf. [14]):

$$\Delta t_{i+1/2} = \frac{h_{i+1/2}}{\max(d_i^{II} - w_{i+1} \ , \ -d_{i+1}^{I} - w_i)} \ , \tag{2.16}$$

where d_i^{II} and d_{i+1}^{I} are the extreme right and left wave speeds at the points x_i and x_{i+1}, respectively, obtained by solving the Riemann problem; w_i is the velocity of the node x_i, i.e., the slope of the intercell boundary. The condition (2.16) means that we estimate the time within which the left characteristic (in linearized analysis, this is a straight line), emanating from the i+1-th node, achieves the i-th node, as well as the time within which the right characteristic, emanating from the i-th node, achieves the i+1-th node. From these two time steps we take the minimal one.

The Courant coefficient c_{cfl} (or coefficient of reserve; see, e.g., [14]) is a correction to the nonlinearity of the PDE system. To calculate the velocity w_i on one hand it is necessary to know the time step Δt, and on the other hand w_i participates in the determination of Δt. By this reason at time level n+1 we use Δt obtained at the preceding level n. The coefficient c_{cfl}<1 (usually 0.5 to 0.9) may be corrected during the computation. If the grid moves with high velocity, then it may be necessary to decrease c_{cfl} greatly to ensure the stability.

In linearized analysis on the fixed grid, we use a sound approach to get $\Delta t_{i+1/2}$

$$\Delta t_{i+1/2} = \frac{h_{i+1/2}}{\max(u_{i+1/2} + a_{i+1/2} \ , \ a_{i+1/2} - u_{i+1/2})} \tag{2.17}$$

where $a_{i+1/2}$ is the sound speed. The formula (2.17) can be also used if the mesh-moving speed is not too large.

In 2D the choice of the admissible Δt may be estimated to the t-hyperbolic by Friedrichs' system [14]. The system of equations (2.2) in the differential form is reduced to the following:

$$\tilde{A}(\boldsymbol{f})\frac{\partial \boldsymbol{f}}{\partial t} + \tilde{B}(\boldsymbol{f})\frac{\partial \boldsymbol{f}}{\partial x} + \tilde{C}(\boldsymbol{f})\frac{\partial \boldsymbol{f}}{\partial y} = 0,$$

where$\tilde{A}$,$\tilde{B}$,$\tilde{C}$ are $n\times n$ symmetric matrices and$\tilde{A}$ is positive defined. The above system is linearized as follows:

$$A\frac{\partial \boldsymbol{f}}{\partial t}+B\frac{\partial \boldsymbol{f}}{\partial x}+C\frac{\partial \boldsymbol{f}}{\partial y}=0,$$

where A,B,C are constant matrices. The energy form to this system is

$$\frac{\partial}{\partial t}\left(A\boldsymbol{f},\boldsymbol{f}\right)+\frac{\partial}{\partial t}\left(B\boldsymbol{f},\boldsymbol{f}\right)+\frac{\partial}{\partial y}\left(C\boldsymbol{f},\boldsymbol{f}\right)=0.$$

It can be shown that after discretizing the above system on the rectangular mesh with steps h_x, h_y, when solving the Cauchy problem, the energy norm of the discrete function $\boldsymbol{f}^{n+1}_{i+1/2,j+1/2}$ does not increase; i.e., the scheme is stable in the following sense:

$$\sum_{i,j=-\infty}^{\infty}\left(A\boldsymbol{f}^{n+1}_{i+1/2,j+1/2},\boldsymbol{f}^{n+1}_{i+1/2,j+1/2}\right)\leq\sum_{i,j=-\infty}^{\infty}\left(A\boldsymbol{f}^{n}_{i+1/2,j+1/2},\boldsymbol{f}^{n}_{i+1/2,j+1/2}\right)$$

subject to

$$\Delta t\left(\frac{1}{\Delta t_x}+\frac{1}{\Delta t_y}\right)\leq 1 \tag{2.18}$$

where $\Delta t_x,\Delta t_y$ are admissible 1D time steps in the x and y directions, respectively. The formula (2.18) has been proved on the parallelogram mesh in [14].

In the case of the curvilinear mesh, to calculate the admissible time step in the $(i{+}1/2,j{+}1/2)$-th cell, the formula (2.18) is generalized as follows [14]:

$$\Delta t_{i+1/2,j+1/2}=\frac{\Delta t^{'}\,\Delta t^{''}}{\Delta t^{'}+\Delta t^{''}}, \tag{2.19}$$

where

$$\begin{aligned}
\Delta t^{'} &= \frac{h^{'}}{\max(d^{II}_{41}-w_{41};-d^{I}_{23}-w_{23})},\\
\Delta t^{''} &= \frac{h^{''}}{\max(d^{II}_{12}-w_{12};-d^{I}_{34}-w_{34})},\\
h^{'} &= \frac{A_{1234}}{0.5\sqrt{(x_4+x_3-x_1-x_2)^2+(y_4+y_3-y_1-y_2)^2}},\\
h^{''} &= \frac{A_{1234}}{0.5\sqrt{(x_3+x_2-x_4-x_1)^2+(y_3+y_2-y_4-y_1)^2}}.
\end{aligned}$$

Here, $\Delta t^{'}$ and $\Delta t^{''}$ are the admissible time steps for 1D scheme in the ξ and η direction, respectively, $h^{'},h^{''}$ are the "average heights" of the bottom face A_{1234}; w is the velocity of the corresponding cell edge in the normal direction to this edge. For example, w_{12} is the velocity of the edge 12 in the normal direction determined by (2.14). Moreover, d^{II}_{12} and d^{II}_{41} are the so-called "extreme right wave" speeds defined from solving the Riemann problem to the faces $122'1'$ and $11'4'4$, respectively; d^{I}_{23} and d^{I}_{34} are the "extreme left wave" speeds to the faces $233'2'$ and $433'4'$, respectively.

The resulting time step over all mesh is given by

$$\Delta t=c_{cfl}\,\min_{i,j}\,\Delta t_{i+1/2,j+1/2}.$$

The GLFC scheme is of second-order accuracy in time and space in the domains of smooth flow provided that the mesh is close to rectangular and the spacing in the ξ and η directions satisfy

$$h_{i+1/2} - h_{i-1/2} = \mathcal{O}((h_{i+1/2})^2)\ , \quad h_{j+1/2} - h_{j-1/2} = \mathcal{O}((h_{j+1/2})^2),$$

and the nodes-motion speed is small

$$x_{i,j}^{n+1} - x_{i,j}^{n} = \mathcal{O}((h_{i+1/2})^2)\ , \quad x_{i,j}^{n+1} - x_{i,j}^{n} = \mathcal{O}((h_{j+1/2})^2)\ .$$

We close this subsection by pointing out that the values $\boldsymbol{f}^{n+1}$ are obtained directly on the moving mesh, which is in contrast to the approach of Tang et al. [27, 36] where interpolations are used on the updated mesh.

2.7. GLFC scheme in 1D. We now briefly describe the GLFC scheme in 1D case. The gas flow is governed by the following quasilinear hyperbolic system:

$$\frac{\partial \boldsymbol{\sigma}}{\partial t} + \frac{\partial \boldsymbol{a}}{\partial x} = 0, \tag{2.20}$$

where the vector-valued functions are $\boldsymbol{\sigma}=(\rho, \rho u, E)^\top$ and $\boldsymbol{a}=(\rho u, p+\rho u^2, u(E+p))^\top$. For the smooth flow, the system (2.20) is equivalent to the system of integral equations

$$\oint_C \boldsymbol{\sigma} dx - \boldsymbol{a} dt = 0. \tag{2.21}$$

We consider the cell of the moving mesh depicted in FIG. 2.2 (here instead the variable ξ we use x). After integrating (2.21) along the contour C of the cell, we obtain the system of the finite-difference equations

$$\boldsymbol{\sigma}_{i+1/2}^{n+1} h_{i+1/2}^{n+1} - \boldsymbol{\sigma}_{i+1/2}^{n} h_{i+1/2}^{n} - \boldsymbol{\sigma}_{i+1}^{n+1/2} h_{i+1} + \boldsymbol{\sigma}_{i}^{n+1/2} h_i + \Delta t(\boldsymbol{a}_{i+1}^{n+1/2} - \boldsymbol{a}_{i}^{n+1/2}) = 0, \tag{2.22}$$

where $h_{i+1/2}^n = x_{i+1}^n - x_i^n$, $h_i = x_i^{n+1} - x_i^n$. We update the cell average values $\boldsymbol{f}_{i+1/2}^{n+1}$ by following the two-step procedure described for the 2D scheme. At the first stage, predictor, assuming piecewise linear distribution of the flow parameters along x, we compute the intermediate values $\bar{\boldsymbol{f}}_{i+1/2}^{n+1}$ via (2.22). We utilize $\boldsymbol{f}_{1,i+1/2}^{n}$ and $\boldsymbol{f}_{2,i+1/2}^{n}$ via (2.7) by applying the monotonicity algorithm. At the second stage, corrector, after computing prewave values $\boldsymbol{f}_{2,i-1/2}^{n+1/2}$, $\boldsymbol{f}_{1,i+1/2}^{n+1/2}$ via (2.12), we solve the Riemann problem by taking into account the velocity of nodes and then determine the postwave values $\boldsymbol{f}_{i}^{n+1/2}$. Substituting them into (2.22) we obtain $\boldsymbol{f}_{i+1/2}^{n+1}$.

3. Grid Generation

To generate a regular adaptive-harmonic mesh we formulate the problem to minimize harmonic (Dirichlet's) functional written for a surface [25, 26, 8, 18]. The relevant notations are shown in FIG. 3.1.

Let us consider the functional defining the adaptive-harmonic grid clustered in regions of large gradients of the function $f(x, y)$:

$$I = \iint \frac{((1+f_x^2)(x_\xi^2 + x_\eta^2) + 2 f_x f_y (x_\xi y_\xi + x_\eta y_\eta) + (1+f_y^2)(y_\xi^2 + y_\eta^2)}{(x_\xi y_\eta - x_\eta y_\xi)\sqrt{1 + f_x^2 + f_y^2}} d\xi d\eta. \tag{3.1}$$

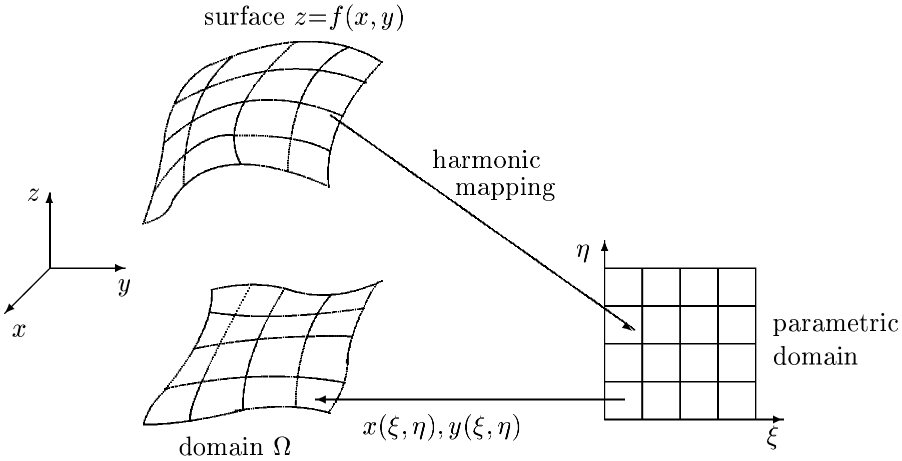

FIG. 3.1. *Harmonic coordinates on the surface of the graph of a control function $z=f(x,y)$.*

The function f is usually referred to as a control/(monitor) function. The problem of grid adaptation is formulated as follows. Let the coordinates of the grid nodes be given, which is formed by quadrilateral elements. The problem is to find new coordinates of the grid nodes minimizing the functional (3.1) values computed for a mapping of the unit square onto every grid cell. In unsteady problems such a formulation for grid adaptation is considered at every time step.

In order to control the number of grid nodes in the layer of large gradients, it has been suggested to multiply the control function f by a coefficient c_a; i.e., replace f by $c_a f$ in (3.1); see [5, 18, 27, 40]. In general, the larger the coefficient of adaptation c_a is, the stronger grid lines condensation in the layer of large gradients we get.

If the control function f is discontinuous then, when adapting, the cell size might approach zero in the vicinity of the discontinuity. In [3] it has been shown that the discrete functional in the 1D case has the infinite barrier preventing the grid cells from degenerating. In the 2D case, despite the infinite barrier for some c_a disappearing, the iterative procedure of minimizing the functional prevents grid lines overlap. This allows for obtaining highly condensed grid lines in the vicinity of discontinuities.

In practical computation of 2D unsteady flows, redundant gridline compression leads to very small time steps. Moreover, we have to reduce c_a in order not to leave the admissible set of convex grids (when all quadrilateral cells are convex). This makes the mesh stop "to feel" the discontinuities of mild and low intensity and reacts only to discontinuities of large intensity. Thus, sometimes (mainly in 2D unsteady flows) it is necessary to smooth the discontinuous control function f in regions of large gradients. There are several ways to do it. In [29, 27, 36] the monitor function is smoothed in the vicinity of discontinuity. We use another approach by introducing an additional parameter restricting the gradient rise for the control function. In our approach the number of grid points in the vicinity of the shock is defined by the flow solver property to smear the shocks. The second-order Godunov's type scheme smears shocks within 2 to 3 cells, which is independent of the cell thickness [39]. In fact, it is not necessary to place in the shock zone more than 2 points, since the shock zone is of infinitely small thickness. This is in contrast to the boundary or internal layers in viscous flows where real physical gradients of the functions exist and require precise description. Thus, in the hyperbolic problems, the coefficient c_a plays the following role: When increasing c_a the mesh begins "to feel" the discontinuities of less intensity; when decreasing the sensitivity of the mesh is also decreased. In the limit of c_a=0 we obtain a quasiuniform harmonic mesh independent of the control function.

Thus, the coefficient c_a is only responsible for selecting those discontinuities of the function $f(x, y)$ to which the mesh is adapted. Therefore, it is necessary to have one more controlling parameter determining the range of the gradients that approximates the discontinuity line in the solution and simultaneously restricts maximum gradient of the grid control function. It is accomplished as follows. First, at the mesh nodes we define the function $\tilde{f} = c_a f^h$, where f^h is an interpolant of f whoes values in the nodes equal the value of f. Then we specify D_{max}, maximal value of modulus of gradient to the control function f, say $D_{max}=\lambda \max(|\nabla\tilde{f}|)$, where the coefficient $\lambda<1$ and $|\nabla\tilde{f}|=\sqrt{\tilde{f}_x^2+\tilde{f}_y^2}$. Next the gradient of the function is updated via

$$\nabla\tilde{f}^* = \begin{cases} D_{max}\nabla\tilde{f}/|\nabla\tilde{f}| & \text{if } |\nabla\tilde{f}| > D_{max} \ , \\ \nabla\tilde{f} & \text{otherwise} \ . \end{cases} \tag{3.2}$$

Then the resulting values of $\tilde{f}_x^*$ and $\tilde{f}_y^*$ are substituted into (3.1) to replace f_x and f_y.

It is not necessary to use this procedure to simulate 1D and steady 2D flows. In the case of rapidly developed unsteady supersonic flows, the above procedure should be applied to prevent the cells to be folded in the vicinity of the shock waves. The question of mesh adaptation to the vector-function, i.e., using several control functions, has been investigated in [1, 19].

In 1D, to generate the inverse harmonic mapping it is required to minimize the following functional [25, 26]

$$I = \int_0^1 \frac{1}{x_\xi\sqrt{1+c_a^2 f_x^2}}d\xi. \tag{3.3}$$

Here we use $c_a f$ instead of f, which indicates that we seek for the arc-length equidistribution in the metric of the curve $c_a f$.

4. Minimization of the Functional

In this section, numerical implementations about minimizing the functional (3.1) and (3.3) will be described.

4.1. 1D Case. Using the midpoint rule gives the discretized functional for (3.3):

$$I^h = \sum_{i=1}^{i_{max}} \frac{\Delta\xi}{(x_\xi)_{i+1/2}\sqrt{1+c_a^2(f_x)_{i+1/2}^2}}, \tag{4.1}$$

where i_{max} is the number of intervals in space, step $\Delta\xi$ is set equal to 1, and the underlying derivatives are computed via

$$(x_\xi)_{i+1/2} = (x_{i+1}-x_i)/\Delta\xi \quad , \qquad (f_x)_{i+1/2} = (f_{i+1}-f_i)/(x_{i+1}-x_i).$$

To minimize (4.1), Newton's method is applied:

$$x_i^{p+1} = x_i^p - \tau \frac{\partial I^h}{\partial x_i} \left[\frac{\partial^2 I^h}{\partial x_i^2} \right]^{-1}, \tag{4.2}$$

where the iterative parameter $\tau \leq 1$, the first and second derivatives are defined by

$$\frac{\partial I^h}{\partial x_i} = \frac{-1}{(x_i - x_{i-1})^2 \sqrt{1 + c_a^2 (f_x)_{i-1/2}^2}} + \frac{1}{(x_{i+1} - x_i)^2 \sqrt{1 + c_a^2 (f_x)_{i+1/2}^2}},$$

$$\frac{\partial^2 I^h}{\partial x_i^2} = \frac{2}{(x_i - x_{i-1})^3 \sqrt{1 + c_a^2 (f_x)_{i-1/2}^2}} + \frac{2}{(x_{i+1} - x_i)^3 \sqrt{1 + c_a^2 (f_x)_{i+1/2}^2}}.$$

When deriving the above derivatives we fix metric, i.e. $(f_x)_{i+1/2}$, which guarantees the infinite barrier property for any $c_a > 0$ and prevents grid lines from overlapping [3].

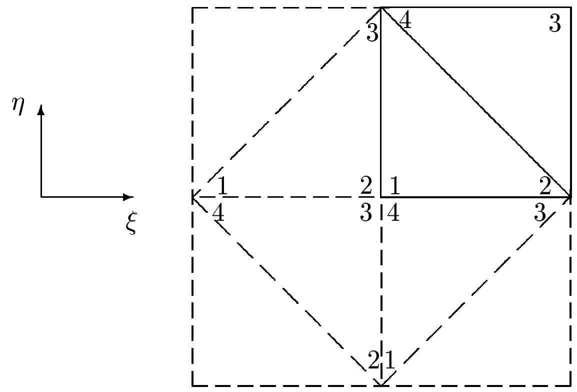

FIG. 4.1. *Computing cell is a unit square in parametric plane. Solid lines indicate the cell considered, dash lines show the adjacent cells.*

4.2. 2D Case. Approximation of the functional (3.1) is performed on the mesh of quadrilaterals, each of them is a unit square in the logic plane (see FIG. 4.1):

$$I^h = \sum_{i=1}^{i_{max}} \sum_{k=1}^{4} \frac{1}{4} [F_k]_i, \tag{4.3}$$

where i_{max} is the total number of grid cells, and k indicates the vertex number in a cell. In (4.3), F_k is defined by

$$F_k = \frac{D_1[1 + (f_x)_k^2] + D_2[1 + (f_y)_k^2] + 2D_3 (f_x)_k (f_y)_k}{J_k [1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \tag{4.4}$$

where

$$\begin{aligned}
D_1 &= (x_{k+1}-x_k)^2+(x_{k-1}-x_k)^2,\\
D_2 &= (y_{k+1}-y_k)^2+(y_{k-1}-y_k)^2,\\
D_3 &= (x_{k+1}-x_k)(y_{k+1}-y_k)+(x_{k-1}-x_k)(y_{k-1}-y_k),\\
J_k &= (x_{k+1}-x_k)(y_{k-1}-y_k)-(x_{k-1}-x_k)(y_{k+1}-y_k).
\end{aligned}$$

If the set of convex grids is not empty, then the system of the following algebraic equations has at least one solution being the convex mesh:

$$R_x=\frac{\partial I^h}{\partial x_i}=0,\quad R_y=\frac{\partial I^h}{\partial y_i}=0.$$

Consider the method of minimizing the functional (4.3) by assuming the grid to be convex at the p-th step of the iterative procedure. We use the quasi-Newton method in the sense that the Hessian is a diagonal matrix:

$$\begin{aligned}
x_i^{p+1} &= x_i^p-\tau\left(R_x\frac{\partial R_y}{\partial y_i}-R_y\frac{\partial R_x}{\partial y_n}\right)\left(\frac{\partial R_x}{\partial x_i}\frac{\partial R_y}{\partial y_i}-\frac{\partial R_y}{\partial x_i}\frac{\partial R_x}{\partial y_i}\right)^{-1},\\
y_i^{p+1} &= y_i^p-\tau\left(R_y\frac{\partial R_x}{\partial x_i}-R_x\frac{\partial R_y}{\partial x_i}\right)\left(\frac{\partial R_x}{\partial x_i}\frac{\partial R_y}{\partial y_i}-\frac{\partial R_y}{\partial x_i}\frac{\partial R_x}{\partial y_i}\right)^{-1}.
\end{aligned}\tag{4.5}$$

To demonstrate how to find R_x, R_y and their derivatives, we consider one of four triangles into which the cell is divided by two diagonals, e.g., with vertex 1, 2, and 4, see FIG. 4.1. Integrand (4.4) is independent of the rotation of the coordinates ξ,η, and as a result the formulas below hold for the remaining 3 triangles. For the triangle 124 setting k=1, k+1=2, k−1=4 in (4.4) gives

$$\begin{aligned}
D_1 &= (x_2-x_1)^2+(x_4-x_1)^2,\\
D_2 &= (y_2-y_1)^2+(y_4-y_1)^2,\\
D_3 &= (x_2-x_1)(y_2-y_1)+(x_4-x_1)(y_4-y_1),\\
J_1 &= (x_2-x_1)(y_4-y_1)-(x_4-x_1)(y_2-y_1).
\end{aligned}\tag{4.6}$$

We introduce notation $F_1=U/V$, where

$$\begin{aligned}
&U=D_1\alpha+D_2\beta+D_3\gamma,\quad V=J_1,\quad \alpha=\frac{\left[1+(f_x)_1^2\right]}{M_1},\\
&\beta=\frac{\left[1+(f_y)_1^2\right]}{M_1},\quad \gamma=\frac{(f_x)_1(f_y)_1}{M_1},\quad M_1=\sqrt{1+(f_x)_1^2+(f_y)_1^2}\,.
\end{aligned}\tag{4.7}$$

For convenience we omit the subscript of F_1, i.e., using F to indicate it. Using the chain rule for F=U/V gives

$$\begin{aligned}
&F_x=(U_x-FV_x)/V,\qquad F_y=(U_y-FV_y)/V,\\
&F_{xx}=(U_{xx}-2F_xV_x-FV_{xx})/V,\qquad F_{yy}=(U_{yy}-2F_yV_y-FV_{yy})/V,\\
&F_{xy}=F_{yx}=(U_{xy}-F_xV_y-F_yV_x-FV_{xy})/V.
\end{aligned}$$

When making differentiation, we assume $f_x,\ f_y$ are independent of x, y. Thus, for the chosen triangle 124 we have

- for vertex 1 of the triangle:

$$\begin{aligned}
&U_x = 2(2x_1 - x_2 - x_4)\alpha + 2(2y_1 - y_2 - y_4)\gamma,\\
&U_y = 2(2y_1 - y_2 - y_4)\beta + 2(2x_1 - x_2 - x_4)\gamma,\\
&U_{xx} = 4\alpha, \quad U_{yy} = 4\beta, \quad U_{xy} = 4\gamma,\\
&V_x = y_2 - y_4, \quad V_y = x_4 - x_2, \quad V_{xx} = V_{xy} = V_{yy} = 0;
\end{aligned}$$

- for vertex 2:

$$\begin{aligned}
&U_x{=}2(x_2 - x_1)\alpha + 2(y_2 - y_1)\gamma,\\
&U_y{=}2(y_2 - y_1)\beta + 2(x_2 - x_1)\gamma,\\
&U_{xx}{=}2\alpha, \quad U_{yy}{=}2\beta, \quad U_{xy}{=}2\gamma,\\
&V_x = y_4 - y_1, \quad V_y = x_1 - x_4, \quad V_{xx} = V_{xy} = V_{yy} = 0;
\end{aligned}$$

- for vertex 4:

$$\begin{aligned}
&U_x{=}2(x_4 - x_1)\alpha + 2(y_4 - y_1)\gamma,\\
&U_y{=}2(y_4 - y_1)\beta + 2(x_4 - x_1)\gamma,\\
&U_{xx}{=}2\alpha, \quad U_{yy}{=}2\beta, \quad U_{xy}{=}2\gamma,\\
&V_x = y_1 - y_2, \quad V_y = x_2 - x_1, \quad V_{xx} = V_{xy} = V_{yy} = 0.
\end{aligned}$$

If the vertex 1 corresponds to the i-th node of the global numeration, then the contributions of four adjacent cells are summarized by

$$\begin{aligned}
&[R_x]_i = \sum_{l=1}^{4}[F_x]_l, \quad [R_y]_i = \sum_{l=1}^{4}[F_y]_l, \quad [R_{xx}]_i = \sum_{l=1}^{4}[F_{xx}]_l,\\
&[R_{xy}]_i = \sum_{l=1}^{4}[F_{xy}]_l, \quad [R_{yy}]_i = \sum_{l=1}^{4}[F_{yy}]_l.
\end{aligned}$$

Here, l=1 corresponds to the triangle 124 of the cell considered, l=2 corresponds to the triangle 231 of the left cell, l=3 corresponds to the triangle 342 of the left-down cell, l=4 corresponds to the triangle 413 of the down cell.

To find the derivatives $(f_x)_i$ and $(f_y)_i$, we discretize the following relations:

$$f_x = (y_\eta f_\xi - y_\xi f_\eta)/J, \quad f_y = (-x_\eta f_\xi + x_\xi f_\eta)/J,$$

and calculate them as average values over the 4 adjacent cells to the *i-th* node

$$[f_x]_i = \sum_{l=1}^{4}[f_x]_l \Bigg/ \sum_{l=1}^{4} J_l, \quad [f_y]_i = \sum_{l=1}^{4}[f_y]_l \Bigg/ \sum_{l=1}^{4} J_l.$$

Here, index l corresponds to the same triangles as described above. For example, if l=1, then for the considered cell we have

$$\begin{aligned}
&[f_x]_1 = [(y_4{-}y_1)(f_2{-}f_1) - (y_2{-}y_1)(f_4{-}f_1)]\,,\\
&[f_y]_1 = [(x_2{-}x_1)(f_4{-}f_2) - (x_4{-}x_1)(f_2{-}f_1)]\,,
\end{aligned}$$

and J_1 is computed via the formula (4.6).

If a flow solver gives the values of the control function in the cell center, it is required to update them to the nodes. Since every i-th node (except boundary nodes) is surrounded by 4 cells, it can be done by

$$f_i = c_a \sum_{l=1}^{4} f_l^c J_l \Big/ \sum_{l=1}^{4} J_l, \tag{4.8}$$

where i is a global node number, f_l^c is the value in the cell center, c_a is the coefficient of adaptation which may depend on the node position, i.e. $c_a{=}c_a(x,y)$. Finally, it is convenient to normalize the control function:

$$\tilde{f}_i = \frac{f_i - f_{min}}{f_{max} - f_{min}} \sqrt{(x_{mx} - x_{min})^2 + (y_{max} - y_{min})^2}.$$

The above formulas can be used for generating the quasiuniform mesh in the physical domain Ω by defining $f{=}const$ and, therefore, $f_x{=}f_y{=}0$. The matter of untangling the initial prepared folded mesh has been considered in [8, 18].

4.3. Redistribution of the Boundary Nodes.

The algorithm of redistributing the boundary nodes, consisting of using constrained minimization, has been suggested in [3]. Such an approach leads to consistent redistribution of the grid nodes inside the domain Ω and on its boundary $\partial\Omega$, which increases the reliability of grid generation and modeling the flow problem. In this approach we minimize the following functional

$$\tilde{I}^h = \sum_{i=1}^{i_{max}} \sum_{k=1}^{4} \frac{1}{4} [F_k]_i + \sum_{l \in \mathcal{L}} \lambda_l G_l = I^h + \sum_{l \in \mathcal{L}} \lambda_l G_l, \tag{4.9}$$

where the constraints $G_l{=}G(x_l, y_l){=}0$ define on $\partial\Omega$, λ_l are the Lagrange multipliers, $\mathcal{L}$ is the set of the boundary nodes. The function $G(x,y)$ is assumed piecewise differentiable.

If the set of convex grids is not empty, the system of algebraic equations has at least one solution being the convex mesh:

$$R_x = \frac{\partial I^h}{\partial x_i} + \lambda_i \frac{\partial G_i}{\partial x_i} = 0, \qquad R_y = \frac{\partial I^h}{\partial y_i} + \lambda_i \frac{\partial G_i}{\partial y_i} = 0, \qquad G_i = 0. \tag{4.10}$$

Here $\lambda_i{=}0$ if $i \notin \mathcal{L}$, and constraints are defined to the boundary nodes $i \in \mathcal{L}$. The method of solving the system (4.10) is described in [3].

5. Numerical Results

With the preparations in the above sections, we are now ready to summarize our moving mesh algorithm. At each time step, the numerical scheme for solving 1D or 2D equations of gas dynamics contains the following stages:

1. Generate the mesh at the next time level t^{n+1}.

2. Compute the gas dynamics values at time t^{n+1}.

3. Evaluate the derivatives $(f_x)_i$ and $(f_y)_i$ at every mesh node (in 1D case $(f_x)_{i+1/2}$) at t^{n+1}.

4. Make one iteration step and compute the new values of $(x,y)_i$ at t^{n+1} by formulas (4.2) or (4.5).

5. Repeat starting with step 2 to convergence or within given number of iterations.

6. Compute the final gas dynamics values at t^{n+1}.

In this section, two numerical examples are considered: a 1D example consisting of simple waves (without wave interactions) and a 2D example having complicated wave structures. In the 1D case, some detailed error analysis is made.

EXAMPLE 5.1. Consider 1D flow of ideal gas with γ=1.4 and initial parameters [35]:

$$(\rho, u, p) = \begin{cases} (1,0,1)\ , & \text{if } 0\le x < 0.5 \\ (0.125, 0, 0.1)\ , & \text{if } 0.5\le x\le 1. \end{cases}$$

At $t>0$ the wave pattern consists of a rarefaction fan with left and right characteristics $u_{rar}^{lft}\approx-1.18322$ and $u_{rar}^{rght}\approx-0.07027$, a contact discontinuity with $u_{cont}\approx0.92745$, and a shock wave with $u_{sh}\approx1.75216$. We compute this problem up to t=0.25. The boundary conditions at the ends x=0 and 1 do not need to be specified, i.e., we solve the Cauchy problem for the 1D Euler equations.

In the first series of computations, the initial grid is uniform with i_{max}=60. Since by t=0.25 the shock almost approaches the right end x=1, in adaptive computations we add to the right end 10 more segments to exclude the influence of the edge to the nodes being attracted into the shock zone. This treatment does not have any practical effect to the error estimation since only 3 to 7 of them enter into the control interval $0\le x\le 1$ when adapting. The admissible time step $\triangle t$ is calculated from the stability condition (2.15), with c_{cfl}=0.5 for the fixed mesh and c_{cfl}=0.6 for the moving mesh. At n=1 the time step is obtained from the linearized condition (2.17) on the fixed mesh for both the fixed and moving mesh cases.

The error of calculation is estimated for density using the numerical solution ρ^h and exact ρ^{ex} in the L_1 norm via the formula

$$\begin{aligned} ||Er||_{L_1} &= \int_0^1 |\rho^h(x) - \rho^{ex}(x)|dx \\ &\approx \sum_{i=1}^{i_{max}} \sum_{j=1}^{j_{max}} \left|\rho_{i+1/2} + q_j h_{i+1/2}\rho_{i+1/2}^{(1)} - \rho^{ex}(x_{i+1/2} + q_j h_{i+1/2})\right| \\ &\quad \cdot\Delta x_{i,j}, \end{aligned} \tag{5.1}$$

where each zone $(i, i{+}1)$ is partitioned uniformly into j_{max}=100 segments, which implies $q_j=(j-0.5)/j_{max}-0.5$ and $\Delta x_{i,j}=h_{i+1/2}/j_{max}$. In (5.1) the derivative of ρ is computed using a linear interpolation

$$\rho_{i+1/2}^{(1)} = \frac{\rho_{i+3/2} - \rho_{i+1/2}}{h_{i+3/2} + h_{i+1/2}} + \frac{\rho_{i+1/2} - \rho_{i-1/2}}{h_{i+1/2} + h_{i-1/2}}\ .$$

The control function f in the functional (4.1) used for this computation is the density ρ since it is sensitive to both the shock and contact discontinuity. With Newton's method (4.2), we perform p_{iter}=50 iterations at each time step. It is found that using the indicator

$$\max_i \left|x_i^{p+1} - x_i^p\right| < \delta,$$

where $\delta \approx 10^{-5}$ to 10^{-4}, to stop the Newton iterations gives lower accuracy. To comply with the stability condition within the first 5 time steps, we set p_{iter}=5 since we begin calculation on the uniform mesh. The computed density is displayed in FIG. 5.1, from which we observe that the moving mesh solution with c_a=8 provides better resolution of all structures in the wave pattern. In FIG. 5.1d a small oscillations (≈1.8%), observed for the fixed mesh solution, disappears on the moving one. In Table 5.1, the numerical errors with different choices of the degree of adaptation, i.e. c_a, are presented. To identify the influence of adaptation on different waves, we calculate the errors gained throughout the rarefaction wave, contact discontinuity, and shock, respectively:

$$\|Er\|_{L_1}^{rar} = \int_0^{x_a} |\rho^h - \rho^{ex}| dx, \quad \|Er\|_{L_1}^{cont} = \int_{x_a}^{x_b} |\rho^h - \rho^{ex}| dx,$$

$$\|Er\|_{L_1}^{sh} = \int_{x_b}^{1} |\rho^h - \rho^{ex}| dx,$$

where $x_a = 0.5(u_{rar}^{rght} + u_{cont})t, \quad x_b = 0.5(u_{cont} + u_{sh})t$, which are also presented in Table 5.1.

We observe that the shock is smeared over two cells on both meshes, see FIG. 5.1d, but since the spacings in the shock zone are reduced by about factor of 8 and due to suppressing oscillations, the error $\|Er\|_{L_1}^{sh}$ is decreased by a factor of 9.67; see the cases c_a=0 and c_a=8 in Table 5.1. The error $\|Er\|_{L_1}^{rar}$ is decreased by a factor of 2.44 since the moving mesh solution more precisely describes the weak discontinuity zones where the left and right characteristics of the rarefaction fan join the flow with constant parameters. Moreover, in the domain of the rarefaction wave, the spacings on the moving mesh are twice smaller than on the fixed one; see FIG. 5.1b. Thickness of the contact discontinuity is also reduced, see FIG. 5.1c, and adaptation gives the winning in accuracy by factor of 3.11. Note that the contact zone, when treated by the second-order scheme, is smeared with time (on the uniform mesh proportionally to $t^{1/3}$ [39]), and the mesh adaptation cannot prevent this process from happening. Thus, as expected, the maximal gain of the accuracy is provided by the shock zone.

One can compare the above results with the moving mesh results presented in [29] where this problem was computed using the moving mesh PDE technique. Different monitor functions on the moving mesh have allowed the accuracy gaining by factor of 1.09 to 1.81 in comparison with the fixed mesh solution.

To compare the computer costs, the number of mesh iterations at every time step p_{iter} should be multiplied by the ratio of admissible time steps on the moving and fixed mesh; see TABLE 5.1. We observe that when c_a=8, we have to use approximately 238 times ($p_{iter} \cdot n(c_a = 8)/n(c_a = 0) = 50 \cdot 319/67 \approx 238$) more computer time than that for the fixed mesh computation. Obviously this cannot be justified by the gaining of accuracy of factor 3.32. This disadvantage of the moving mesh approach is in general true in 1D computations. However, our numerical experience shows that in 2D with strong gridline compressions, we need to perform only 1 or 2 mesh iterations at each time step. Nevertheless the problem of decreasing the running time is to be solved, since stability requires the time step Δt to be small enough.

With the purpose to decrease computer costs, we have used the following numerical strategy. Up to time t_1 (t_1 was varied from 0.15 to 0.22) we first performed at every time step p_{iter}=5 mesh iterations, and then in the interval $(t_1, 0.25)$ we defined p_{iter}=50. Increasing p_{iter} from 5 to 50 can be executed gradually within several time

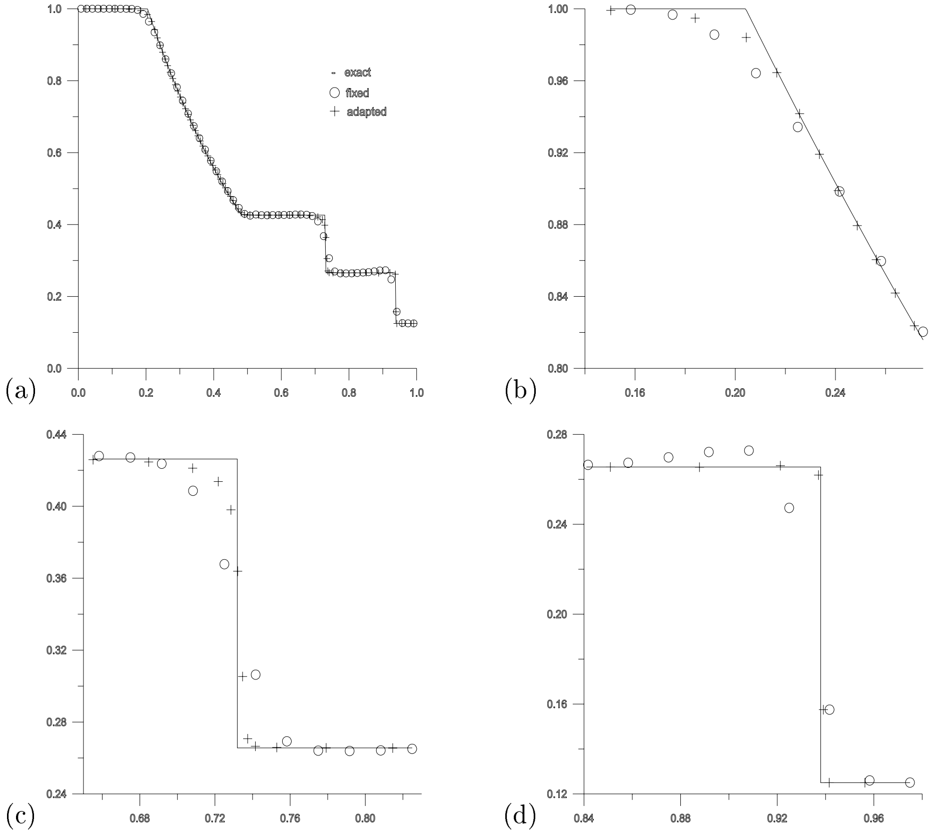

FIG. 5.1. *Example 5.1. (a) Plots of density; (b) Close-up of solution in the vicinity of rarefaction fan; (c) contact discontinuity; and (d) shock.*

steps. Results turned out to be unsatisfactory. In all cases the error was increased even in comparison with the fixed mesh solution.

In the second series of computations, the grid consists of i_{max}=120 intervals. Results are also presented in TABLE 5.1. We obtain the gaining in accuracy by a factor of 2.63. As in the preceding case, the maximal gain by a factor of 11.72 is obtained to the shock zone and minimal by a factor of 1.60 to the rarefaction fan zone.

EXAMPLE 5.2. As a 2D example, we consider a spherical explosion between two parallel walls at z=0 and z=1. Initially the gas is at rest with parameters $(p, \rho)_{out}$=$(1, 1)$ everywhere except in a sphere centered at $(0, 0, 0.4)$ with radius 0.2. Inside the sphere $(p, \rho)_{in}$=$(5, 1)$, γ=1.4. This problem was proposed by LeVeque [23].

The initial jump in pressure results in an outward moving shock, a contact discontinuity, and an inward-moving rarefaction wave. There occurs interactions among these waves and between the waves and the walls. The flow by t=0.7 consists of several shocks and strong contact/(tangential) discontinuity surrounding the low density region near the center.

First, until the initial outward shock reaches the wall z=0, the solution is spheri-

TABLE 5.1. *Comparison of fixed* (c_a=0) *and moving grid solutions for shock tube problem. Here,* n *is the number of time steps,* τ *is the iterative parameter in procedure (4.2).*

c_a	n	τ	$\|Er\|_{L_1}$	$\|Er\|^{rar}_{L_1}$	$\|Er\|^{cont}_{L_1}$	$\|Er\|^{sh}_{L_1}$
mesh $i_{max}=60$						
0	67	–	0.005926	0.002229	0.002256	0.001441
1	127	0.6	0.004125	0.001529	0.001723	0.000874
2	179	0.6	0.002950	0.001002	0.001340	0.000608
4	236	0.35	0.002443	0.001220	0.000975	0.000249
8	319	0.35	0.001787	0.000913	0.000725	0.000149
mesh $i_{max}=120$						
0	135	–	0.003159	0.001058	0.001339	0.000762
1	324	0.7	0.001564	0.000671	0.000740	0.000153
2	469	0.6	0.001330	0.000598	0.000576	0.000157
3	577	0.5	0.001199	0.000663	0.000471	0.000065

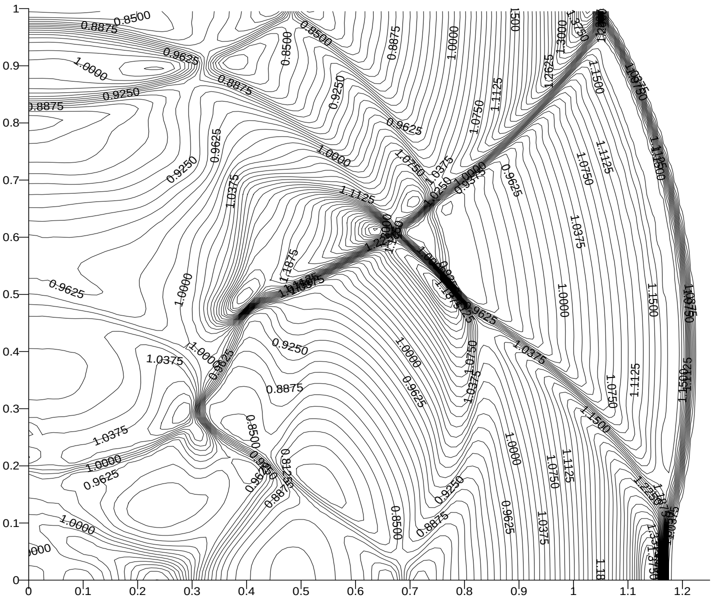

FIG. 5.2. *Pressure contours from* 0.775 *to* 1.5 *with increment* 0.0125 *in the x-z plane at* t=0.7 *computed on the rectangular mesh* 100×140.

cally symmetric. It remains cylindrically symmetric, and we use the 2D axi-symmetric formulation for the problem; see system (2.2) where we use the variable z instead x and x instead y and parameter ν=1 in (2.1) for determining the vector $\boldsymbol{c}$. In the half plane with constant ϕ (where ϕ is an angle in cylindrical coordinates) we compute the domain $(x,z)\in[0,1.4]\times[0,1]$. On the axis x=0 we define the symmetric boundary conditions, on the walls z=0, 1 the reflection condition is employed, and on x=1.4 undisturbed values $(u,v,p,\rho)_{out}$ are used. The fixed mesh is rectangular. In all computations the coefficient c_{cfl} is chosen as 0.6. The pressure contours at t=0.7 computed on the fixed mesh 100×140 are presented in FIG. 5.2.

We perform adaptation by minimizing the discrete functional (4.3). On the walls

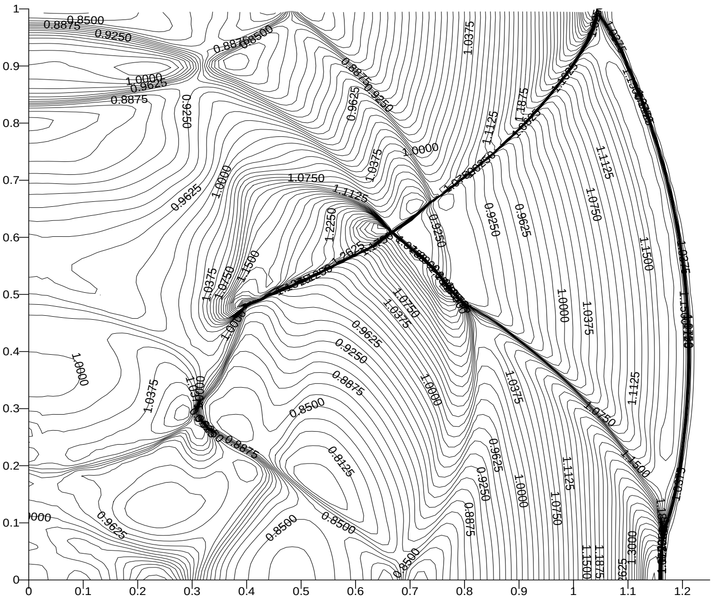

Fig. 5.3. *Pressure contours computed on the adapted mesh* 100×140.

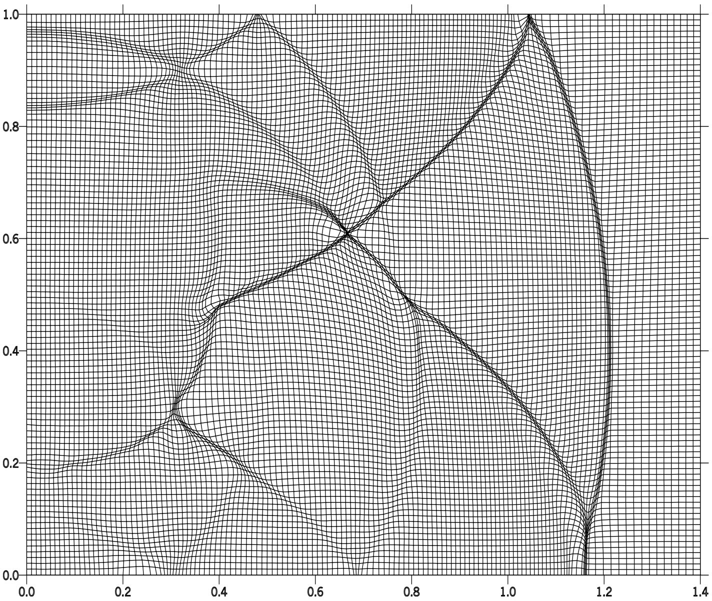

Fig. 5.4. *Adapted mesh* 100×140*, p is a control function,* c_a=0.21*,* D_{max}=5.5.

and the axis of symmetry, we apply constrained minimization by solving the system (4.10). The adaptive procedure is switched on at t=0.45. Since strong gridline compression results in decreasing the admissible time step, i.e., Δt falls down proportionally to the increase of the cell aspect ratio, we use the following strategy. In the main part of the time interval, the computation is performed on not very condensed mesh, and at some moment t'=0.7$-\delta t$ we attain a maximum compression of gridlines

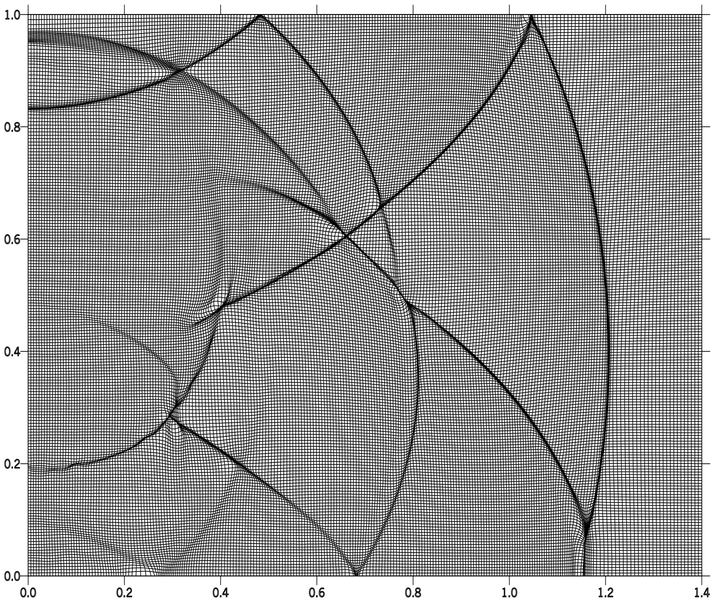

FIG. 5.5. *Adapted mesh* 200×250*, p is a control function,* c_a=0.18*,* D_{max}=5.

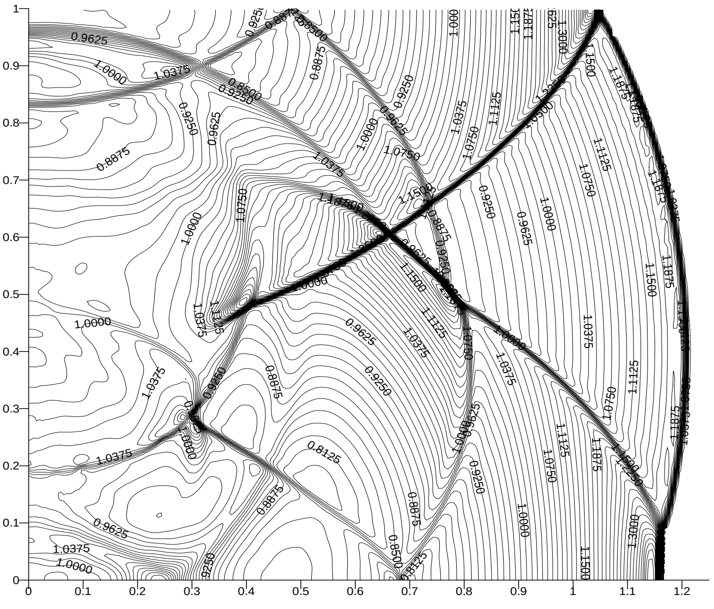

FIG. 5.6. *Pressure contours computed on the rectangular mesh* 200×250.

in the vicinity of discontinuities. Following this methodology, we set a rather small coefficient of adaptation, c_a=0.07 to 0.1, in the time interval $(0.45, 0.65)$. In this time interval, it is then sufficient to perform 3 mesh iterations at every time step. The iterative parameter τ in quasi-Newton procedure (4.5) is set to 0.9. The admissible time step Δt is decreased by 5 to 10 times in comparison with Δt on the rectangular mesh.

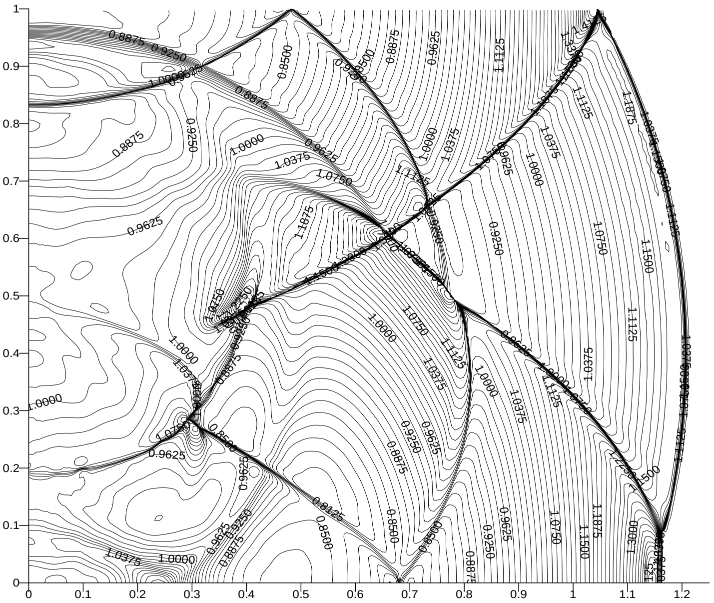

FIG. 5.7. *Pressure contours computed on the adapted mesh* 200×250.

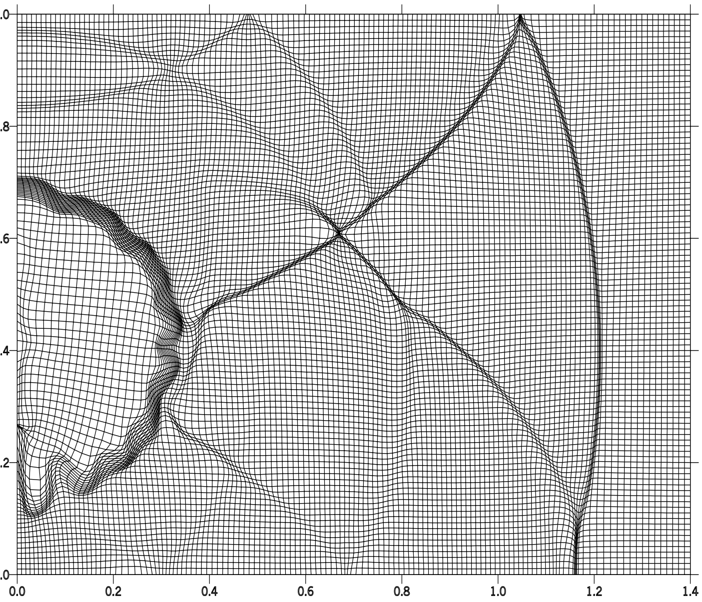

FIG. 5.8. *Adapted mesh* 100×140, *ρ is a control function,* c_a=0.4, D_{max}=6.

In the final interval $(0.65, 0.7)$, the coefficient c_a is increased to the range $(0.18, 0.4)$, and then time step falls down to 5% to 0.5% of that for the rectangular mesh. The time step is not a constant since the mesh constantly "breaths" and accordingly the value of Δt periodically decreases and then increases. At this stage it is sufficient to perform 1 to 2 mesh iterations at every time step. The maximal

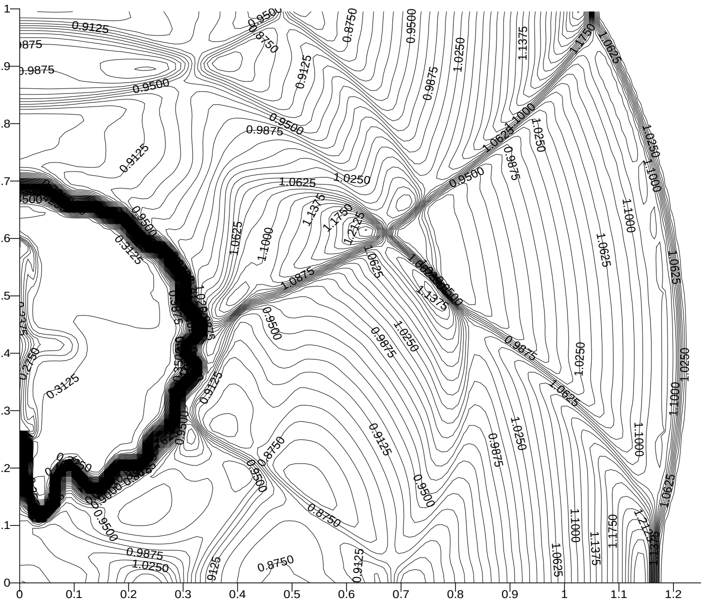

FIG. 5.9. *Density contours from* 0.2 *to* 1.3 *with increment* 0.0125 *computed on the rectangular mesh* 100×140.

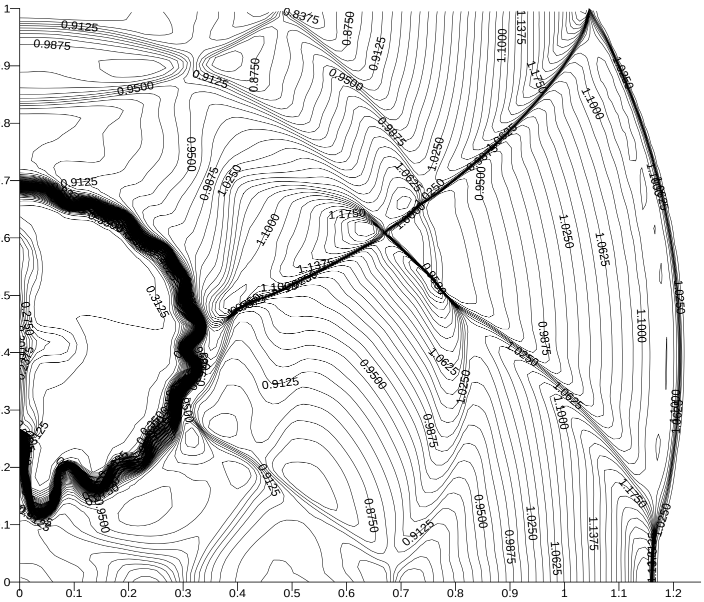

FIG. 5.10. *Density contours computed on the adapted mesh* 100×140.

gradient of the control function in (3.2) is set to D_{max}=5 to 6. Note that the larger the parameter D_{max} is used, the thinner the smeared shock can be obtained. For steady flows we do not use the restriction procedure (3.2) at all [3]; and in some cases of unsteady flow with simple shock structure, the parameter D_{max} used is about 100 [19]. However, for the present modeling it is not appropriate to use large D_{max} due to the complicated structure of many interacting shocks.

In the first series of adaptive computations the pressure p is used as a control function. In FIGS. 5.3 and 5.4 the pressure contours and adapted mesh are shown, respectively. We see that the shock thickness is decreased significantly in comparison with the fixed mesh solution presented in FIG. 5.2. However, the gridlines can be strongly condensed only to the outer shock and two more ones adjoining it. This is due to insufficient number of grid points. In the next calculation we use the mesh 200×250; see FIG. 5.5. Here, both the intensive shocks and compression waves are well depicted by the condensed grid lines. The pressure contours, computed on the rectangular and adapted meshes, are presented in FIGS. 5.6 and 5.7, respectively.

Since p is continuous across the tangential discontinuity, the mesh does not "feel" it. In the next calculation, the density ρ is used as the control function. It is seen from FIG. 5.8 that the mesh 100×140 resolves the tangential discontinuity very well. The density contours computed on the quasiuniform and adaptive meshes are presented in FIGS. 5.9 and 5.10, respectively. It is observed that, in spite of the fact that in FIG. 5.8 c_a is twice larger than that in FIG. 5.4, the shocks are less well indicated by the grid lines. This occurs due to the influence of the contact discontinuity, which takes away quite large amount of the grid points when the density is chosen as the control function.

6. Concluding Remarks

The computational results presented in this work show that the method of adaptive-harmonic grid generation can be successfully applied to unsteady problems of gas dynamics. The grid is clustered in such a way that we can obtain satisfactory resolutions of the discontinuities without using a large number of grid nodes.

In the 2D case, the adapted mesh looks like a set of blocks with the boundaries defined automatically by the condensed gridlines. Inside every "block," the mesh is quasiuniform, and the solution is smooth. This approach may be considered as a further development of the moving block technique presented in the monograph of Godunov et al. [14], where the flow domain is divided into blocks with boundaries being the shock waves or contact discontinuities. Advantage of the moving adaptive grid approach is that such a "cutting" is performed automatically. In real-world computations to obtain such a complex nonstationary configuration of the blocks, even if we would be able to capture their boundaries "by hand," we would have serious problems joining those blocks.

At a first glance, it seems that for unsteady problems the grid adaptation increases the total time of computation significantly in our 2D calculations approximately by factor of 10 to 20 due to additional updating the flow parameters after every mesh iteration and reduction of the admissible time step. A detail analysis removes this impression. Suppose we want to reduce the thickness of shock wave smearing by a factor of 20. With the moving mesh method, we attain such a cell aspect ratio in the shock zone. However, with the rectangular mesh, we need to increase number of grid points by 20 times in not only x and y directions but also the time direction, since the time step falls as well. As a result we need computer memory about 400 times larger, and the running time will increase by a factor of 20^3=8000. With the moving mesh approach, the computer cost is approximately the number of iterations at each time step multiplying by the ratio of the admissible time step. So in this case the running time increases by a factor of 20β (in reality by a lesser factor since we perform calculations with the smallest Δt only within a part of time t), where β is the number of iterations at each time step which is about 2 to 3 in the 2D computations. This comparison demonstrates the great saving in computer memory and gaining in computer time for the use of the proposed moving mesh methods. The advantage of the adaptive procedure seems more impressive in 3D calculations.

Acknowledgments. This work was supported by Hong Kong Research Grants Council (CERG HKBU 2044/00P and 2083/01P) and Russian Fund of Fundamental Research (project code 02-01-00236).

REFERENCES

[1] B.N. Azarenok and S.A. Ivanenko, *Application of adaptive grids in numerical analysis of time-dependent problems in gas dynamics.* Comput. Maths. Math. Phys., 40:1330–1349, 2000.

[2] B.N. Azarenok, *Realization of a second-order Godunov's scheme.* Comput. Math. in Appl. Mech. and Engin., 189:1031–1052, 2000.

[3] B.N. Azarenok, *Variational barrier method of adaptive grid generation in hyperbolic problems of gas dynamics.* SIAM J. Numer. Anal., 40:651–682, 2002.

[4] M. Ben-Artzi and J. Falcovitcz, *A second order Godunov-type scheme for compressible fluid dynamics.* J. Comput. Phys. 55:1–32, 1984.

[5] W.M. Cao, W.Z. Huang, and R.D. Russell, *An r-adaptive finite element method based upon moving mesh PDEs.* J. Comput. Phys. 149:221–244, 1999.

[6] H.D. Ceniceros and T.Y. Hou, *An efficient dynamically adaptive mesh for potentially singular solutions.* J. Comput. Phys., 172:609–639, 2001.

[7] A.A. Charakhch'yan and S.A. Ivanenko, *Curvilinear grids of convex quadrilaterals.* USSR Comput. Math. Math. Phys. 28:126–133, 1988.

[8] A.A. Charakhch'yan and S.A. Ivanenko, *A variational form of the Winslow grid generator.* J. Comput. Phys. 136:385–398, 1997.

[9] P. Colella and P.R. Woodward, *The piecewise parabolic method (PMM) for gas-dynamical simulations.* J. Comput. Phys., 54:174–201, 1984.

[10] R. Courant and K.O. Friedrichs, *Supersonic Flow and Shock Waves.* Springer-Verlag, 1985.

[11] J.K. Dukowicz, *Efficient volume computation for three-dimensional hexahedral cells.* J. Comput. Phys., 74:493–496, 1988.

[12] A.S. Dvinsky, *Adaptive grid generation from harmonic maps on Riemannian manifolds.* J. Comput. Phys. 95:450–476, 1991.

[13] S.K. Godunov, *A finite difference method for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics.* Mat. Sb., 47:357–393, 1959.

[14] S.K. Godunov, et al(eds.), *Numerical Solution of Multidimensional Problems in Gas Dynamics.* (in Russian). Nauka Press, Moscow, 1976.

[15] A. Harten and J.M. Hyman, *Self–adjusting grid methods for one–dimensional hyperbolic conservation laws.* J. Comput. Phys. 50:235–269, 1983.

[16] C.L. Ingram and D.S. McRay, *Extension of a dynamic solution-adaptive grid algorithm and solver to general structured milti-block configurations.* AIAA 96-0294, AIAA 34th Aerospace Science Meeting, Reno, NV, Jan. 1996.

[17] S.A. Ivanenko, *Adaptive grids and grids on surfaces.* Comp. Maths. Math. Phys, 33:1179–1193, 1993.

[18] S.A. Ivanenko, *Harmonic Mappings, Chapt. 8 in: Handbook of Grid Generation.* J.F. Thompson et al. (eds.), CRC Press, Boca Raton, Fl, 1999.

[19] S.A. Ivanenko and B.N. Azarenok, *Application of moving adaptive grids for numerical solution of nonstationary problems in gas dynamics.* Int. J. for Numer. Meth. in Fluids, 39:1–22, 2002. (See also http://www.math.ntnu.no/conservation/2001/043.html).

[20] P. Knupp, L. Margolin, and M. Shashkov, *Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods.* J. Comput. Phys., 176:93–128, 2002.

[21] V.P. Kolgan, *Application of the principle of minimal derivatives towards constructing finite difference schemes to compute discontinuous solutions in gas dynamics.* Scientific Notes of Central Aerohydrodynamical Institute (in Russian), Moscow, 3:68–77, 1972.

[22] A.G. Kulikovskii, N.V. Pogorelov, and A.Yu. Semenov, *Mathematical Aspects of Numerical Solution of Hyperbolic Systems.* Monographs and Surveys in Pure and Applied Mathematics 188, Chapman and Hall/CRC, Boca Raton, FL, 2001.

[23] J.O. Langseth and R.J. LeVeque, *A wave propagation method for 3D hyperbolic conservation laws.* J. Comput. Phys. 165:126–166, 2000.

[24] F. Liu, S. Ji, and G. Liao, *An adaptive grid method and its application to steady Euler flow calculations.* SIAM J. Sci. Comput. 20:811–825, 1998.

[25] V.D. Liseikin, *On generation of regular grids on n-dimensional surfaces.* USSR Comput. Math. Math. Phys. 31:47–57, 1991.

[26] V.D. Liseikin, *Grid Generation Methods.* Springer-Verlag: New York, 1999.

[27] R. Li, T. Tang, and P. Zhang, *Moving mesh methods in multiple dimensions based on harmonic maps.* J. Comput. Phys. 170:562–588, 2001.

[28] R. Li, T. Tang, and P. Zhang, *A moving mesh finite element algorithm for singular problems in two and three space dimensions.* J. Comput. Phys., 177:365–393, 2002.

[29] J.A. Mackenzie, R.D. Russell, and J.M. Stockie, *A moving mesh method for one dimensional hyperbolic conservation law.* SIAM J. Sci. Comput. 22:1791–1813, 2000.

[30] D.S. McRay and R. Laflin, *Dynamic Grid Adaptation and Grid Quality.* Chapt. 34 in Handbook of Grid Generation. (CRC Press, Boca Raton, Fl, 1999).

[31] A.V. Rodionov, *Increase of accuracy of the Godunov's scheme.* USSR Computational Math. and Math. Phys., 27:1853–1860, 1987.

[32] B.L. Rozhdestvenskii and N.N. Yanenko, *Systems of Quasilinear Equations and Their Applications to Gas Dynamics.* Amer. Math. Soc. Transl. of Math. Monographs, 55, AMS, Providence, RI, 1983.

[33] J.S. Sampson, *Some properties and applications of harmonic mappings.* Ann. École Norm. Sup. 11:211–228, 1978.

[34] R. Schoen and S.T. Yau, *On univalent harmonic maps between surfaces*, Invent. Math. 44:265–278, 1978.

[35] G.A. Sod, *A survey of finite difference methods for the systems of nonlinear hyperbolic conservation laws.* J. Comput. Phys. 27:1–31, 1978.

[36] H.Z. Tang and T. Tang, *Moving mesh methods for one– and two–dimensional hyperbolic conservation laws.* http://www.math.hkbu.edu.hk/~ttang. To appear in SIAM J. Numer. Anal.

[37] J.F. Thompson, *A survey of dynamically-adaptive grids in the numerical solution of partial differential equations.* Appl. Numer. Maths, 1:3–27, 1985.

[38] O.V. Ushakova, *Conditions of nondegeneracy of three-dimensional cells.* A formula of a volume of cells. SIAM J. Sci. Comput., 23:1273–1289, 2001.

[39] E.V. Vorozhtsov and N.N. Yanenko, *Methods for Localization of Singularities in Numerical Solution of Gasdynamics Problems.* Springer Series in Computational Physics, 1990.

[40] W. Ren and X.P. Wang, *An iterative grid redistribution method for singular problems in multiple dimensions.* J. Comput. Phys., 159:246–273, 2002.

[41] P.A. Zegeling, *Moving Grid Techniques.* 37 in Handbook of Grid Generation. CRC Press, Boca Raton, Fl, 1999.