JOURNAL OF COMPUTATIONAL PHYSICS **136,** 385–398 (1997)
ARTICLE NO. CP975750

A Variational Form of the Winslow Grid Generator

A. A. Charakhch'yan and S. A. Ivanenko

Computing Center of the Russian Academy of Sciences, Vavilov Str. 40, Moscow GSP-1, 117967 Russia
E-mail: chara@ccas.ru

Received December 5, 1996; revised May 12, 1997

A structured grid generator based on a new approximation of Dirichlet's functional is developed. An unconstrained minimization process guarantees that all quadrilateral grid cells are convex at each iteration. Numerical results are presented in comparison with those for the original form of the Winslow method. A generalization to the case of adaptive grids based on harmonic maps between surfaces is considered.

INTRODUCTION

The problem of constructing two-dimensional grids will be considered in the following standard formulation. A grid G must be constructed for a domain Ω in the (x, y) plane with given coordinates of the boundary nodes $(x, y)_{i1}$, $(x, y)_{im}$, $(x, y)_{1j}$, $(x, y)_{nj}$ such that

$$G = \{(x, y)_{ij}, i = 1, ..., n; j = 1, ..., m\}.$$

There are many approaches to the problem (see [1], for example). If the given configuration of the boundary nodes is such that coordinate lines of the grid must not be bent strongly, then as a rule all algorithms produce quite satisfactory grids. This is not the case if the coordinate lines must be bent considerably to generate a satisfactory grid. In extreme cases most algorithms begin to generate grids which contain self-intersecting cells and are therefore unsuitable for computations. For many applications, such a difficulty can be overcome by constructing block grids when the grid at each block has its own structure (see [2] and a number of papers from [3], for example). Even so, fixed block grids are often unsuitable for moving block interfaces or boundaries. For example, we consider a problem of grid generation encountered when simulating cumulative jet propagation into a conical target which is portrayed schematically in Fig. 1a. The dashed lines show positions of an interface between two substances at different times. A typical domain for grid generation is illustrated in Fig. 1b. Clearly, the domain can be easily divided into two blocks in such a way that it is straightforward to generate a grid in each block. Nevertheless, a fixed block approach is unsuitable for computations because, at different times the dividing line (shown dashed in Fig. 1b) will be connected with different boundary nodes. It then becomes necessary to use a remapping procedure between grids of different structures, perhaps at every time step.

The Winslow method [4] and its different generalizations for differential and variational formulation have been considered in many papers [1, 2, 5–17]. Direct variational methods based on the minimization of particular discrete functionals are proposed in [18, 19]. Unlike the preceding methods the present algorithm guarantees the convexity of all quadrilateral grid cells practically for any realistic distortion of boundary lines while the number of grid nodes and the structure of the grid remain fixed. Other elliptic grid generation methods give such a guarantee only if the number of grid nodes tends to infinity [10, 29]. An algorithm which overcomes this drawback has been proposed [20], and is based on a constrained minimization of a finite-difference function with Lagrangian multipliers as additional independent variables. Herein, an alternative algorithm which solves the unconstrained minimization problem for a function dependent only on the coordinates of the grid nodes is described. As a result, the computational cost of the algorithm is comparable with that for other elliptic grid generation methods.

All grid cells are convex at each iteration of the minimization process. This is very important in time-dependent problems with moving boundaries when the number of iterations cannot be large because the grid must be constructed at each time step.

The unconstrained minimization algorithm [21, 22] has been used in the simulation of cumulative jets [23, 24] and flows in basins [25] with very complicated boundaries for about 10 years. This paper presents a detailed account of the unconstrained minimization algorithm, and its generalization to the case of adaptive grids.

The method is described in detail in Section 1. Numerical results are discussed in Section 2. Examples from computations of real-world problems are presented, where the original Winslow method generates grids with self-intersecting cells while the method presented generates quite satisfactory grids consisting only of convex cells. An adaptive grid generator with the same property is described in Section

0021-9991/97 $25.00

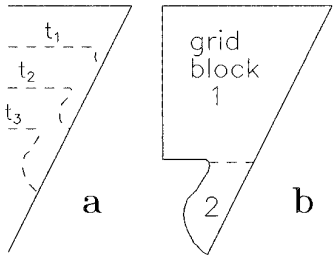

FIG. 1. Cumulative jet propagation into a conical target. (a) Positions of the interface between two substances (dashed line) at different times $t_3 > t_2 > t_1$; (b) a typical grid generation region for the upper substance.

3. The high reliability of an adaptive grid generator is the necessary condition for its successful application since the grid lines must also be bent strongly to adjust the singularities of numerical solutions.

1. CONVEX GRID GENERATOR

The problem of grid generation can be treated as a discrete analog of the problem of finding functions $x(\xi, \eta)$ and $y(\xi, \eta)$, producing one-to-one mapping of the parametric square

$$0 < \xi < 1, \quad 0 < \eta < 1$$

onto a domain Ω.

Instead of the parametric square on the plane ξ, η the parametric rectangle is often introduced to simplify the computational formulas

$$1 < \xi < n, \quad 1 < \eta < m.$$

This rectangle is associated with the square grid (ξ_i, η_j) on the plane ξ, η such that $\xi_i = i$, $\eta_j = j$, $i = 1, ..., n$; $j = 1, ..., m$.

It can be shown that if a smooth mapping of one domain onto another with one-to-one mapping between boundaries possesses a positive Jacobian, then such a mapping will be one-to-one.

Hence, the curvilinear coordinate system constructed in domain Ω will be non-degenerate if the Jacobian of the mapping $x(\xi, \eta)$, $y(\xi, \eta)$ is positive:

$$J = x_\xi y_\eta - x_\eta y_\xi > 0. \tag{1}$$

Thus, the problem of constructing the curvilinear coordinates in the domain Ω can be formulated as the problem of finding the smooth mapping of a parametric square onto the domain Ω, which satisfies the condition of the Jacobian positiveness. The mapping between boundaries must be one-to-one.

Consequently, in the discrete case for the grid G the discrete analog of the Jacobian positiveness must be also imposed. However, it is not obvious how to approximate inequalities (1). For example, this question is discussed in [10] where it is shown that in some cases the usual approximations of (1) are unsuitable.

The condition of the grid cell convexity was introduced in [20] as a discrete analog of the Jacobian positiveness. We will follow here the paper [22] where the mapping $x(\xi, \eta), y(\xi, \eta)$ was approximated by quadrilateral finite elements.

Let the coordinates $(x, y)_{i,j}$ of grid nodes be given. To construct the mapping $x^h(\xi, \eta)$, $y^h(\xi, \eta)$ of a parametric rectangle onto the domain Ω such that $x^h(i, j) = x_{i,j}$ and $y^h(i, j) = y_{ij}$ we use quadrilateral isoparametric finite elements [26]. The square cell numbered $i + 1/2, j + 1/2$ on the plane ξ, η is mapped onto the quadrilateral cell on the plane x, y, formed by nodes with coordinates $(x, y)_{i,j}$, $(x, y)_{i,j+1}$, $(x, y)_{i+1,j+1}$, $(x, y)_{i+1,j}$.

The cell vertices are numbered from 1 to 4 in a clockwise direction. The node (i, j) corresponds to vertex 1, node $(i, j + 1)$ to vertex 2, and so on. Each vertex is associated with a triangle: vertex 1 with $\triangle_{412}$, vertex 2 with $\triangle_{123}$, and so on. The doubled area J_k, $k = 1, 2, 3, 4$, of these triangles is introduced as

$$J_k = (x_{k-1} - x_k)(y_{k+1} - y_k) - (y_{k-1} - y_k)(x_{k+1} - x_k),$$

where one should put $k - 1 = 4$ if $k = 1$, $k + 1 = 1$ if $k = 4$.

Functions x^h, y^h for $i \le \xi \le i + 1$, $j \le \eta \le j + 1$ are represented in the form

$$x^h(\xi, \eta) = x_1 + (x_4 - x_1)(\xi - i) + (x_2 - x_1)(\eta - j) + (x_3 - x_4 - x_2 + x_1)(\xi - i)(\eta - j), \tag{2a}$$

$$y^h(\xi, \eta) = y_1 + (y_4 - y_1)(\xi - i) + (y_2 - y_1)(\eta - j) + (y_3 - y_4 - y_2 + y_1)(\xi - i)(\eta - j). \tag{2b}$$

Each side of the square is linearly transformed onto the appropriate side of the quadrilateral. Consequently, the global transformation x^h, y^h is continuous on the cell boundaries. To check the one-to-one property of the transformation (2) we write out the expression for the Jacobian [26]

$$J^h = x^h_\xi y^h_\eta - x^h_\eta y^h_\xi = \det \begin{pmatrix} x_4 - x_1 + A(\eta - j) & x_2 - x_1 + A(\xi - i) \\ y_4 - y_1 + B(\eta - j) & y_2 - y_1 + B(\xi - i) \end{pmatrix},$$

where $A = x_3 - x_4 - x_2 + x_1$, and $B = y_3 - y_4 - y_2 + y_1$. The Jacobian is linear, not bilinear, since the coefficient

before $\xi\eta$ in this determinant is equal to zero. Consequently, if $J^h > 0$ in all corners of the square, it does not vanish inside this square. In corner 1 ($\xi = i$, $\eta = j$) of cell $i + 1/2, j + 1/2$ the Jacobian $J^h(i, j) = (x_4 - x_1)(y_2 - y_1) - (y_4 - y_1)(x_2 - x_1)$, i.e., $J^h(i, j) = J_1$, is the doubled area of the triangle $\triangle_{412}$, introduced above.

From this follows that the condition of the Jacobian positiveness $x^h_\xi y^h_\eta - x^h_\eta y^h_\xi > 0$ is equivalent to the system of inequalities

$$\begin{gathered}[J_k]_{i+1/2,j+1/2} > 0,\\ k = 1, 2, 3, 4; i = 1, ..., n-1; j = 1, ..., m-1.\end{gathered} \tag{3}$$

If conditions (3) are satisfied, then all the grid cells are convex quadrilaterals. The set of grids satisfying these inequalities is called a convex grid set and denoted by D. This set belongs to the Euclidean space R^N, where $N = 2(n-2)(m-2)$ is the total number of degrees of freedom of the grid equal to double the number of its internal nodes.

Finally the problem is formulated as follows. The convex grid, satisfying inequalities (3), must be constructed in the domain Ω for the given coordinates of the boundary nodes.

The method of grid generation guaranteeing the one-to-one mapping on the continuous level was proposed by Winslow [4]. Two families of grid lines are constructed as contours of harmonic functions $\xi(x, y)$, $\eta(x, y)$ satisfying two Laplace's equations

$$\Delta\xi = 0, \quad \Delta\eta = 0,$$

with Dirichlet's boundary conditions. After transforming to independent variables ξ, η, these equations take the form

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0, \quad \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0, \tag{4}$$

where $\alpha = x^2_\eta + y^2_\eta$, $\beta = x_\xi x_\eta + y_\xi y_\eta$, and $\gamma = x^2_\xi + y^2_\xi$. The standard approximation of (4) using centered differences for the first-order derivatives will be referred to as the original form of the Winslow method.

Equations (4) are the Euler–Lagrange equations for the functional

$$\begin{aligned} I &= \int [(\nabla\xi)^2 + (\nabla\eta)^2]\, dx\, dy \\ &= \int \frac{x^2_\xi + y^2_\xi + x^2_\eta + y^2_\eta}{J}\, d\xi\, d\eta \end{aligned} \tag{5}$$

used by Brackbill and Saltzman [9]. They added to (5) other functionals and approximated the Euler–Lagrange equations of the total functional by finite differences. Here, we will approximate the functional itself rather than the Euler–Lagrange equations. This approach was also considered by Roache and Steinberg [10], who noted that all usual approximations of (5) ensure a minimally acceptable grid, with no folding, only if $n, m \to \infty$. The presence of the inverse Jacobian in the integrand of (5) should be assumed as the main factor entailing one-to-one mapping, whereas usually this is not taken into account when approximating the Euler equations for the grid generation. The present algorithm is based on a particular approximation whereby the minimum ensures all grid cells to be convex quadrilaterals and guarantees no folding for the given values of n and m. In its implementation the peculiarity of vanishing the Jacobian when the one-to-one property is lost can be used explicitly.

Here we will approximate the functional itself rather than the Euler equations. The mapping $x(\xi, \eta)$, $y(\xi, \eta)$ is approximated by the functions $x^h(\xi, \eta)$, $y^h(\xi, \eta)$ introduced in (2). Substituting these expressions into (5) and replacing integrals over square cells by the quadrature formulas with nodes coinciding with the grid vertices on the plane ξ, η [26], the following discrete analog of (5) can be obtained:

$$I^h = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \sum_{k=1}^{4} \frac{1}{4} [F_k]_{i+1/2,j+1/2}, \tag{6a}$$

where F_k is the integrand evaluated in the kth grid node,

$$\begin{aligned} F_k = [&(x_{k+1} - x_k)^2 + (x_k - x_{k-1})^2 + (y_{k+1} - y_k)^2 \\ &+ (y_k - y_{k-1})^2] J_k^{-1}, \end{aligned} \tag{6b}$$

and J_k is the doubled area of the triangle introduced above.

Consider several properties of the function (6). Recall that the parametric rectangle $1 < \xi < n$, $1 < \eta < m$ was introduced to simplify computational formulas. In order to consider the approximative properties of the function (6) we should introduce the parametric rectangle $0 < \xi < 1$, $0 < \eta < \alpha$, where $\alpha = (m-1)/(n-1)$ is the constant, instead of the unit parametric square as a domain of integration in (5). In this case the continuous limit of the expression $I^h/(n-1)^2$ when $n, m \to \infty$ in such a way that $(m-1)/(n-1) = \alpha = \text{const}$ will be the functional (5).

It is easy to obtain the identity

$$\begin{aligned} I &= \int_0^1 \int_0^\alpha \frac{\begin{array}{r} x^2_\xi + y^2_\xi + x^2_\eta + y^2_\eta - 2(x_\xi y_\eta - x_\eta y_\xi) \\ + 2(x_\xi y_\eta - x_\eta y_\xi) \end{array}}{J}\, d\xi\, d\eta \\ &= \int_0^1 \int_0^\alpha \frac{(x_\xi - y_\eta)^2 + (x_\eta - y_\xi)^2}{J}\, d\xi\, d\eta + 2\alpha. \end{aligned}$$

From this follows that the functional (5) has a lower bound equal to 2α. If this minimum is attained, the mapping is conformal:

$$x_\xi = y_\eta, \quad x_\eta = -y_\xi.$$

To obtain the corresponding property of the discrete analog (6) of the functional (5) consider one term in (6) for $k = 2$. We can assume that $x_2 = 0$ and $y_2 = 0$ since (6b) contains only finite differences of grid node coordinates. In this case we can obtain the similar identity

$$F_2 = \frac{x_1^2 + y_1^2 + x_3^2 + y_3^2}{x_1 y_3 - x_3 y_1} = \frac{\begin{array}{r} x_1^2 + y_1^2 + x_3^2 + y_3^2 - 2(x_1 y_3 - x_3 y_1) \\ + 2(x_1 y_3 - x_3 y_1) \end{array}}{x_1 y_3 - x_3 y_1}$$
$$= \frac{(x_1 - y_3)^2 + (x_3 + y_1)^2}{x_1 y_3 - x_3 y_1} + 2.$$

From this follows that the function $I^h/(n-1)^2$ has on the set D a lower bound equal to $2(m-1)/(n-1)$. If this minimum is attained, coordinates of the grid nodes satisfy the discrete analog of conformal conditions

$$x_1 = y_3, \quad x_3 = -y_1.$$

If these conditions are satisfied for all cells, each grid cell will be a square.

Note that the function (6) is not convex and, in principle, multiple solutions can exist.

Function I^h possesses the following very important property. If $G \to \partial D$ for $G \in D$, where ∂D is the boundary of the set of convex grids D, i.e., if at least one of the quantities J_k tends to zero for some cell while remaining positive, then $I^h(G) \to +\infty$. In fact, suppose that $J_k \to 0$ in (6b) for some cell, but I^h does not tend to $+\infty$. Then the numerator in (6b) must also tend to zero; i.e., the lengths of two sides of the cell tend to zero. Consequently, the areas of all triangles that contain these sides must also tend to zero. Repeating the argument as many times as necessary, we conclude that the lengths of the sides of all grid cells, including those at the boundary of the domain, must tend to zero, which is impossible.

Note that a discrete functional for triangle meshes with the same property was considered in [27].

Thus, if the set D is not empty, the system of algebraic equations

$$R_x = \frac{\partial I^h}{\partial x_{ij}} = 0, \quad R_y = \frac{\partial I^h}{\partial y_{ij}} = 0,$$
$$i = 2, ..., n-1; j = 2, ..., m-1, \tag{7}$$

has at least one solution which is a convex grid. To find it, one must first find a certain initial grid $G_0 \in D$, and then use some method of unconstrained minimization. Since the function (6) has an infinite barrier on the boundary of the set D, each step of the method can be chosen so that the grid always remains convex. Note that in the common case the discrete grid generation equations (7) can have multiple solutions, but we have not met this difficulty in our numerical experimentation.

We first consider a method of minimizing the function (6) assuming that the initial grid $G_0 \in D$ has been found. Suppose that the grid at the lth step of the iterations is determined. Following Brackbill and Saltzman [9], we use the quasi-Newtonian procedure when the $(l+1)$th step is accomplished by solving two linear equations for each interior node

$$\begin{aligned} \tau R_x + \frac{\partial R_x}{\partial x_{i,j}}(x_{i,j}^{l+1} - x_{i,j}^l) + \frac{\partial R_x}{\partial y_{i,j}}(y_{i,j}^{l+1} - y_{i,j}^l) = 0 \\ \tau R_y + \frac{\partial R_y}{\partial x_{i,j}}(x_{i,j}^{l+1} - x_{i,j}^l) + \frac{\partial R_y}{\partial y_{i,j}}(y_{i,j}^{l+1} - y_{i,j}^l) = 0, \end{aligned} \tag{8}$$

where τ is the iteration parameter. Note that (8) is not the Newton–Raphson iteration because only a part of the second derivatives of (6) is taken into account. The rate of convergence for (8) is low by comparison. At the same time the Newton–Raphson method gives a much more complex system of linear equations at each iteration.

Each of the derivatives occurring in (8) is the sum of 12 terms, in accordance with the number of triangles containing the given node as a vertex. Rather than write out such cumbersome expressions, we considered the first and second derivatives of the terms in (6),

$$\frac{\partial F_k}{\partial x_{k-1}} = 2\frac{x_{k-1} - x_k}{J_k} - F_k \frac{y_{k+1} - y_k}{J_k}, \tag{9}$$

and so on. Arrays storing the derivatives of the function (6) were first cleared, and then all grid triangles were scanned and the appropriate derivatives added to the relevant elements of the arrays.

Now we consider an algorithm for the choice of the iteration parameter τ in (8), which was used only for problems with moving boundaries. Recall that the minimized function (6) has an infinite barrier on the boundary of the set of convex grids D. Since the initial grid $G_0 \in D$, the iteration (8) gives, as a rule, a convex grid for any $\tau < 1$. But in extreme cases when G_0 is very close to the boundary of the set D, the grid $G(\tau)$ can cross the boundary of the set in the first iterations (8). Clearly, such condition is fatal for the method because the same barrier on the boundary of the set D does not allow the iterations to return to the set D in the following iterations. To avoid this, a certain basic parameter τ_0 is chosen so that $G(\tau_0/2) \in D$ and $G(\tau_0) \in D$. In the beginning $\tau_0 = 1$. If the above-mentioned conditions are violated, we put $\tau_0 = 1/4$ or $\tau_0 = 1/2$, depending on whether the grids $G(\tau_0/2)$ or $G(\tau_0)$ leave the set D, and so on.

In fixed boundary problems we use the simple choice $\tau = \text{const} \cdot \tau_0$. For time-dependent problems with moving boundaries a version of the method of parabolas was developed. We choose the squared residual of Eqs. (7)

$$W = \sum_{i,j} (R_x^2 + R_y^2)_{i,j}$$

as the controlling quantity. The reason for such choice will be discussed at the end of Section 2. The parabola $W(\tau)$ is constructed from the grids obtained for $\tau = 0$, $\tau = \tau_0/2$, and $\tau = \tau_0$. The parameter τ is then chosen so that $W(\tau) = \min$ in the interval $\theta\tau \le \tau \le \alpha\tau_0$. The parameter $\theta \sim 0.1$ is given a priori and bounds the value of τ away from zero. The parameter α bounds τ above, i.e., prevents a very large extrapolation along the parabola. If $\tau_0 = 1$, i.e., if the boundary of the set D is not crossed, we put $\alpha = 2$. If $\tau_0 < 1$, then $\alpha = 1$. Finally, if the algorithm gives $\tau < \tau_0/2$, the condition $I^h(\tau_0/2) < I^h(0)$ is checked. In cases when this condition is found to be valid, we put $\tau = \tau_0/2$.

For one iteration of the above method a measurement of the computational cost gives the value of about double (but not three times) the cost of the simple iteration. The reason is that the second derivatives of the function (6) are not used in calculating W while they are used in (8) for calculating the direction of minimization. Comparative properties of the simple iterations and the method of parabolas will be discussed for a certain example at the end of Section 2.

In all our computations the iterative process has been observed to have the following properties. If the initial grid is far from the solution of (7), then the grid nodes move quite rapidly during the first iterations and the residual of (7) decreases rapidly. Later on the rate of convergency may fall, but the iterative process may be interrupted before reaching full convergency and the resulting grid will be necessarily convex.

The algorithm described above can be used only if the initial grid is convex. Otherwise, it is necessary either to obtain a convex grid by another algorithm as a preliminary stage of the method or to modify the computational formulas. The first approach is based on the minimization of the function

$$I_D = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \sum_{k=1}^{4} ([\varepsilon - J_k]_{i+1/2,j+1/2})_+^2, \qquad (f)_+ = \max(0, f), \tag{10}$$

for some given $\varepsilon > 0$. This is accomplished by the gradient method with a suitable choice of iteration parameter. The iterative process is broken off as soon as all inequalities (3) are satisfied. This method was used for problems with moving boundaries [23, 24] when the initial interior grid nodes for minimizing (10) were taken from the previous time step. As a result, the initial grid is either already convex or such that a convex grid is obtained after a few iterations.

In fixed boundary problems, the starting grid (generated by another method, for example, by the Winslow method) is often essentially nonconvex, containing numerous self-intersecting cells. In such cases the preliminary stage of the method based on minimizing (10) can be unsuitable. Therefore another approach has been developed [28]. The computational formulas (8) are modified so that the initial grid need not belong to the set of convex grids D. The quantities J_k appearing in the expressions for R_x, R_y and their derivatives are replaced with new quantities $\tilde{J}_k$

$$\tilde{J}_k = \begin{cases} J_k & \text{if } J_k > \varepsilon, \\ \varepsilon & \text{if } J_k \le \varepsilon, \end{cases}$$

where $\varepsilon > 0$ is some sufficiently small quantity.

It is quite important to choose an optimal value of ε so that the convex grid is constructed as fast as possible. The method used for specifying the value of ε is based on the computation of the absolute value of the average area of triangles with negative areas

$$\varepsilon = \max[\alpha S/(N + 0.01), \varepsilon_1],$$

where S is double the absolute value of the total area of triangles with negative areas, and N the number of these triangles. The quantity $\varepsilon_1 > 0$ sets a lower bound on ε to avoid very large values appearing in the computations. The coefficient α is chosen experimentally and is in the range $0.3 \le \alpha \le 0.7$.

In practical implementation, an arbitrary set of grid nodes can be marked as movable during iterations, while all other nodes are considered stationary. All the terms in the function (6) which become independent on movable nodes are excluded from computations. Since the boundary nodes are always marked as stationary, four terms in (6) corresponding to "corner" triangles $\{(1, 2); (1, 1); (2, 1)\}$, $\{(n - 1, 1); (n, 1); (n, 2)\}$, $\{(1, m - 1); (1, m); (2, m)\}$, and $\{(n - 1, m); (n, m); (n, m - 1)\}$ are always excluded from computations. As a result, the method becomes applicable to those domains for which the angle between two intersecting boundaries is greater than or equal to π, despite the fact that the corresponding grid cell becomes nonconvex independent of the positions of the interior nodes (see Fig. 2).

We use two approaches to control the distribution of nodes. The first one was taken from [9] and was used for attracting the grid lines to boundaries (see [22]). Let us consider the function

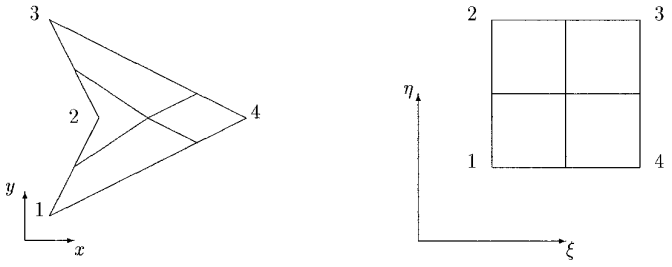

FIG. 2. Example of a domain with such correspondence of boundary nodes that the corner cell is always nonconvex.

$$L^h(G) = I^h(G) + \sum_{i,j} f_{ij} s_{ij},$$

where f_{ij} are the elements of the control array and s_{ij} are the areas of the cells. Clearly, this function possesses the same properties as I^h, i.e., $L^h(G) \to +\infty$ as $G \to \partial D$, $G \in D$. Therefore, the above-described algorithm can be also used for the function $L^h(G)$.

Another approach based on the minimization of a single functional will be considered in Section 3.

2. NUMERICAL EXAMPLES

The unconstrained minimization process presented herein can be treated as a special form of the Winslow method because the algebraic equations (7) are the second-order finite-difference approximations to the Euler–Lagrange equations for the functional (5) (the proof of this fact is omitted). Consequently, results from the unconstrained minimization of the function (6) will be compared with those from the original form of the Winslow method. The numerical algorithm for the latter is taken from [5, 29]. Particular consideration will be given to examples where the original Winslow method generates grids with self-intersecting cells. One may correct these grids with control functions on the right-hand sides of Eqs. (2) [1, 5, 8] or on the left-hand sides by means of a diffusion coefficient [12, 14, 17, 36]. However, it is not a simple task. To demonstrate this, we present one example for the method [8] with control functions evaluated automatically from boundary point distributions.

In general, the original Winslow method is more economical computationally (the computational cost of the present method is approximately seven times the cost of the Winslow method) partly because the number of terms in (6) is almost four times the number of grid nodes and so should be used for meshing until self-intersection cells occur. The present method then becomes very useful for such cases when grid lines must be bent strongly to generate a satisfactory grid.

First we discuss numerical results for a domain from [7] with two deep folds at opposite boundaries. Figure 3 presents the grids generated by the original Winslow method and the Winslow-type method [8] with control functions evaluated automatically from boundary point distributions. It is shown that both grids contain overlapping cells near the corners. The attempt to multiply the control functions from [8] by a factor of 4 turns out to be also unsuccessful (see Fig. 3c). The grid cells near the folds are not improved even when the number of grid nodes is increased four times in each direction.

The present method generates the quite satisfactory grid shown in Fig. 4a. Figure 4b shows an initial grid constructed using straight lines and containing self-intersecting cells. After minimization of the function (10), the grid shown in Fig. 4c was obtained. Although it looks extremely unsatisfactory, it is in fact a convex grid and this is all that is demanded from the initial grid for the basic algorithm of minimization of the function (6).

Figure 5 presents a block of a grid utilized in the simulation of a cumulative jet. The grid generated by the original Winslow method is shown in Fig. 5a. The enlargement shows the section of the grid in which self-intersecting cells occur. Figure 5b shows the grid generated by the present method. It consists only of convex cells.

An example of a domain with a rather complicated boundary (Lake Mendota) is presented in Figs. 6a (the original Winslow method) and 6b (the present method). The enlargement shows one of the sections where the unconstrained minimization method corrects the grid generated by the original method of Winslow. Here an initial grid contains numerous self-intersecting cells. The preliminary minimization of (10) becomes unsuitable while the computational cost of the modified formulas (8) turns out to be quite acceptable.

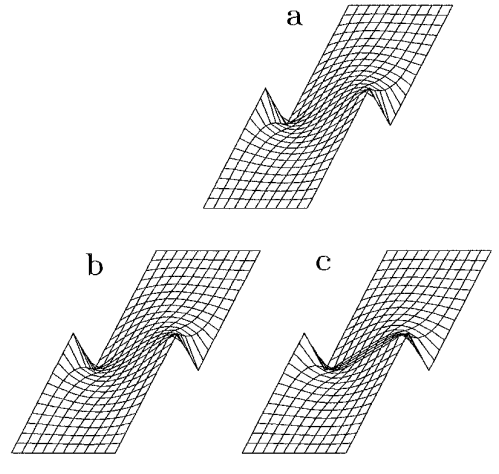

FIG. 3. Folded grids constructed by the Winslow-type methods. (a) The original Winslow method; (b) the method from [8]; (c) the method from [8] with control functions multiplied by a factor of 4.

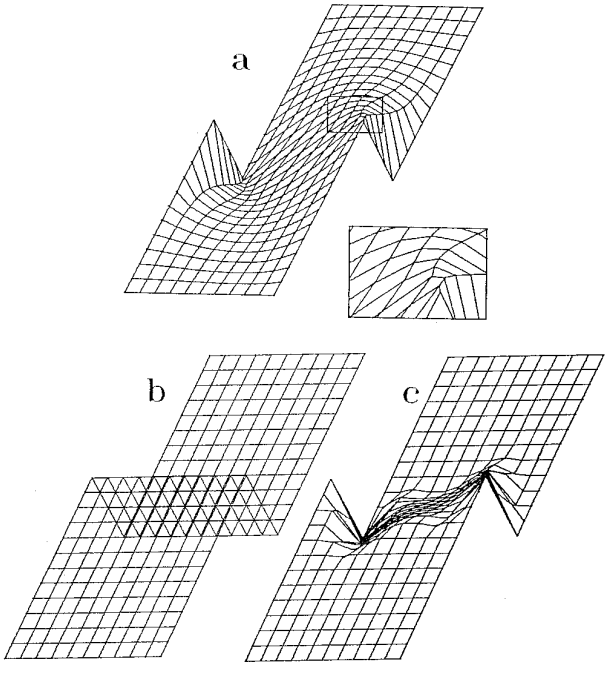

FIG. 4. Grid constructed by the present method. (a) Final grid; (b) initial grid; (c) convex grid after minimization of (10).

The following example was taken from the computation of a high-velocity impact of a thin foil upon a conical target [30]. The original shape of the foil on the radial–axial coordinates is a thin quadrilateral, shown in Fig. 7a. The grid used in the computations has a routine structure with the points A, B, C, and D as corner nodes. Figure 7b shows the shape of the foil within a certain time following the impact. For the domain from Fig. 7b, the present method generates a convex grid which has the same structure as the original grid for the domain from Fig. 7a. Figure 8 shows a fragment of this grid in the vicinity of a point E marked both in this figure and in Fig. 7b.

The last example was also used to compare variants of the method with different choices of the iteration parameter τ. Consider the grid generation technique for the problem in more detail. To construct the grid at each time step, we first determine positions of boundary grid nodes. The interior nodes are calculated by the linear interpolation for the undeformed part of the foil (the left part of the domain in Fig. 7b) while the present method is used only for another part. For the domain shown in Fig. 7b the last part of the grid consists of about 4000 cells. The grid from the previous time step was used to construct the initial grid. As a result the initial grid turns out to be convex.

Besides the minimized function I^h and the total squared residual W, consider the local scaled residual of Eqs. (7)

$$\delta_{ij} = \sqrt{(R_x)_{ij}^2 + (R_y)_{ij}^2}/\sigma_{ij},$$

where σ_{ij} has the same order of magnitude as separate terms occurring in $(R_x)_{ij}$ and $(R_y)_{ij}$. Recall that

$$(R_x)_{ij} = \sum_s (a_s + b_s), \quad (R_y)_{ij} = \sum_s (c_s + d_s),$$

where the subscript s corresponds to triangles containing the node (i, j) as a vertex; a, b, c, and d have the form shown in (9). We use σ_{ij} in the form

$$\sigma_{ij} = \sum_s \sqrt{(|a_s| + |b_s|)^2 + (|c_s| + |d_s|)^2}.$$

The functions I^h, W and $D = \max(\delta_{ij})$ against the iteration number are presented in Fig. 9 both for the simple iterations with $\tau \equiv 0.5$, $\tau \equiv 0.8$ and for the method of parabolas from Section 1. The basic parameter of the method $\tau_0 = 1$ for all iterations. There is no convergence in the case of $\tau \equiv 1$. As one would expect (see Figs. 8a and 8b), the method of parabolas gives somewhat smaller values of I^h and W in comparison with the simple iterations for the same value of the iteration number.

If the function I^h is used as the controlling quantity in the method of parabolas instead of W, the dependence of I^h against the iteration number remains almost the same;

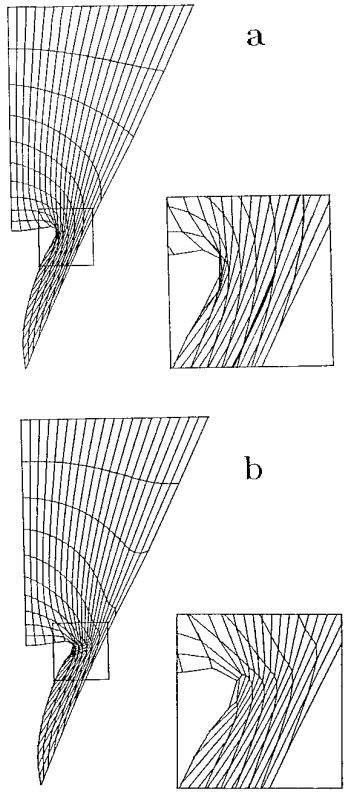

FIG. 5. A grid block for computation of a cumulative jet. (a) The Winslow method; (b) the present method.

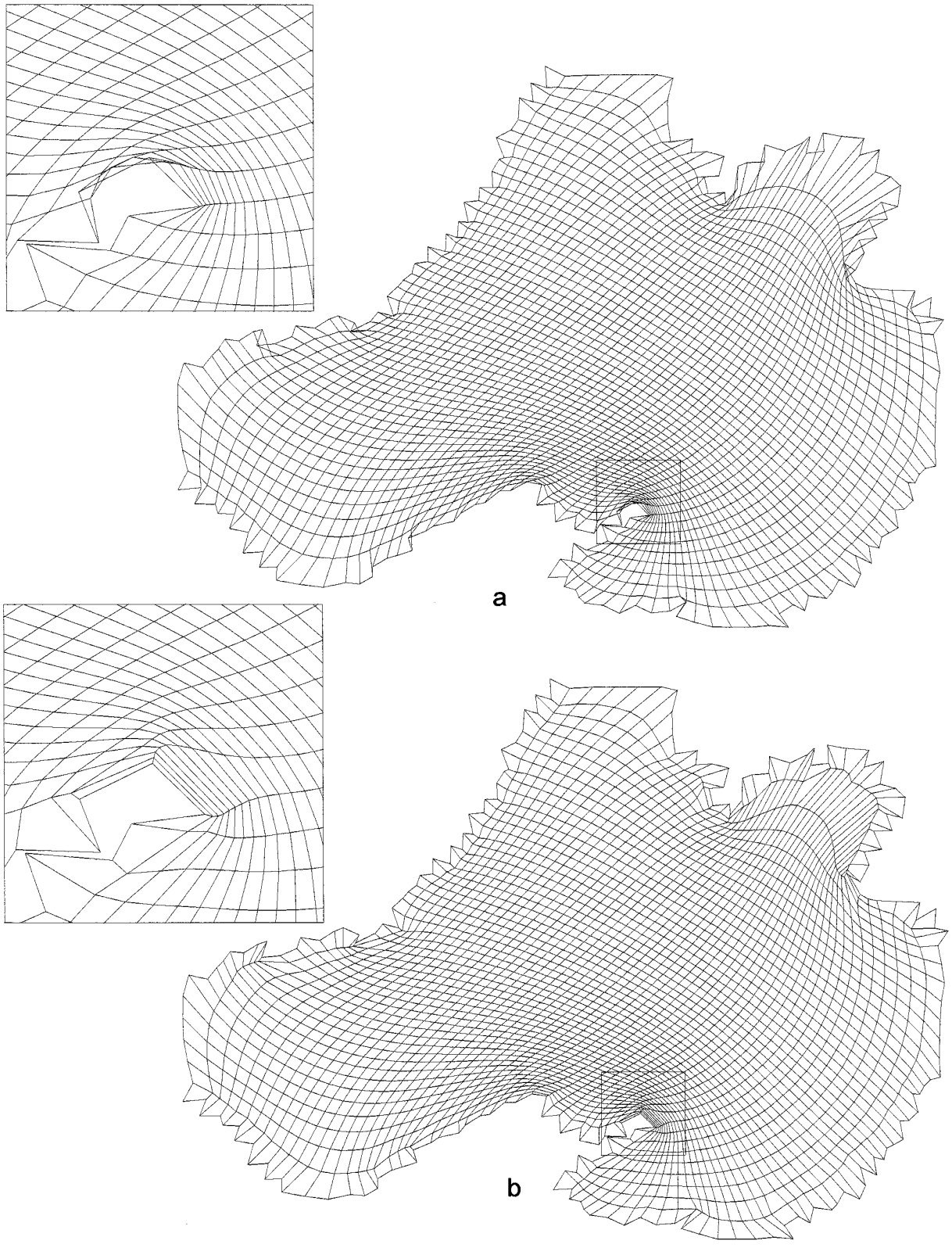

FIG. 6. Grids for Lake Mendota. (a) The Winslow method; (b) the present method.

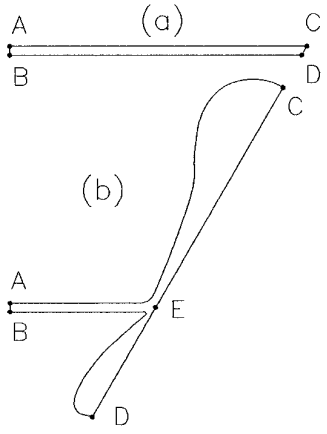

FIG. 7. The domain from computation of a high-velocity impact of a thin foil upon a conical target [30]; A, B, C, and D are the corner grid nodes. (a) The original domain; (b) within a time following the impact.

however, the dependence of W becomes substantially nonmonotonic (see Fig. 9c).

In view of the computational cost of the method of parabolas, the simple iterations with $\tau \equiv 0.8$ turn out to be preferable for decreasing the functions I^h and W down to given values. At the same time the method of parabolas turns out to be preferable for decreasing the function $D = \max(\delta_{ij})$ as it is shown in Fig. 9d. Moreover, the fact of the almost monotonic dependence of D against the iteration number in the case of the method of parabolas can be used to construct a reliable condition to terminate the iterations. Namely, an additional parameter δ is specified, and, if $D \le \delta$, the iteration process is terminated.

3. ADAPTIVE-HARMONIC GRID GENERATOR

Dvinsky [16] was the first to suggest using the theory of harmonic maps for constructing adaptive grids. Brackbill [17] derived an adaptive grid generator from a variational formulation of Winslow's diffusion method combined with a directional control functional. The theory of harmonic maps was used in [17] to define the conditions under which unique solutions of elliptic equations exist. In papers [16, 17] the grid was adapted in the domain of physical space by prescribing a special metric in it.

We apply the theory of harmonic maps to a monitoring surface approach [12] and develop a method for generating quasi-uniform harmonic grids directly on the solution surface. This grid is then projected onto the physical domain and the result is an adaptive-harmonic grid. The first step in this direction was made by Liseikin [31, 33], who obtained a generalization of the functional (5) to the case of surfaces and presented some examples of adaptive grids as numerical solutions of the corresponding Euler–Lagrange equations. In the present paper we obtain the same functional from the theory of harmonic maps (see also [33]).

First we present some common definitions from the survey by Eells and Lemaire [32]. The energy density of a map ϕ: $(M, g) \to (N, h)$ between Riemannian manifolds (surfaces) M and N with metrics g and h is the function $e(\phi)$: $M \to \mathscr{R}(\ge 0)$, defined in local coordinates ξ^i, μ^α as

$$e(\phi)(\xi) = g^{ij}(\xi)\,\frac{\partial \phi\alpha(\xi)}{\partial \xi^i}\,\frac{\partial \phi^\beta(\xi)}{\partial \xi^j}\,h_{\alpha\beta}(\phi(\xi)), \tag{11}$$

where the standard summation convention is assumed, g_{ij} and h_{ij} are the elements of metric tensors g and h on manifolds M and N, and g^{ij} is the inverse metric:

$$g^{ij}g_{jk} = \delta^i_k = \begin{cases} 1 & \text{if } i = k, \\ 0 & \text{if } i \ne k. \end{cases}$$

This means that if g_{ij} are the elements of matrix g, then g^{ij} are the elements of the inverse matrix g^{-1}.

The generalization of Dirichlet's functional is called the energy of ϕ and is defined as

$$\begin{gathered} E(\phi) = \int_M e(\phi)(\xi)\,d\xi, \\ \text{where } d\xi = \sqrt{\det(g)}\,d\xi^1 \cdots d\xi^n. \end{gathered} \tag{12}$$

A smooth map ϕ: $(M, g) \to (N, h)$ is harmonic if it is an extremal of the energy functional E.

Here we consider the case when M is an n-dimensional surface in the Euclidean space $\mathscr{R}^{n+r}$, dim $M = n$, and N is a unit cube $0 < \mu^i < 1$, $i = 1, ..., n$ in $\mathscr{R}^n$. The Euclidean metric in $\mathscr{R}^n$ is $h_{\alpha\beta} = \delta_{\alpha\beta}$. Local coordinates ξ^i and μ^α are the same in this case, and (11) can be simplified to give

$$e(\phi)(\xi) = g^{ij}\,\frac{\partial \xi^\alpha}{\partial \xi^i}\,\frac{\partial \xi^\beta}{\partial \xi^j}\,\delta_{\alpha\beta} = g^{ij}\delta_{ij} = g^{ii} = \mathrm{Tr}(g^{-1}).$$

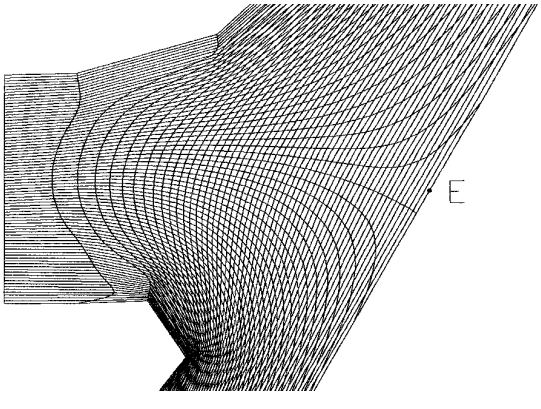

FIG. 8. The fragment of the grid in the vicinity of the point E marked also in Fig. 7b.

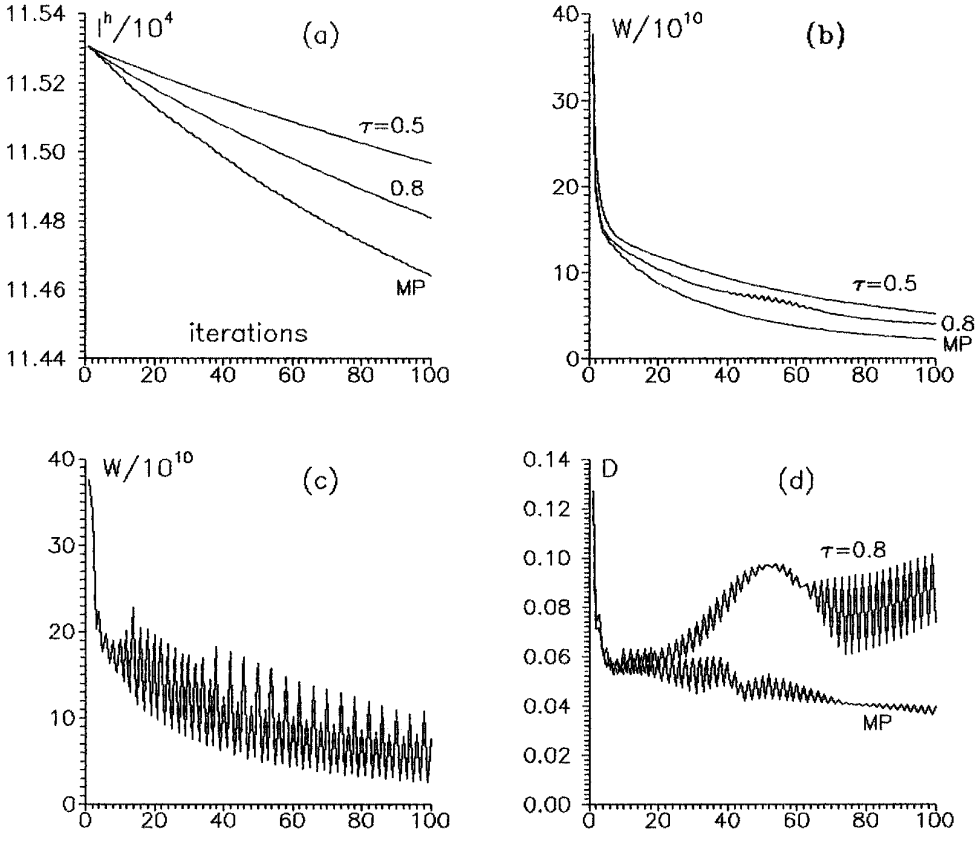

FIG. 9. Functions against the iteration number. (a) I^h: $\tau \equiv 0.5$, $\tau \equiv 0.8$, and the method of parabolas (MP); (b) W: $\tau \equiv 0.5$, $\tau \equiv 0.8$ and MP; (c) W: MP with I^h as the controlling quantity; (d) $D = \max(\delta_{ij})$: $\tau \equiv 0.8$ and MP.

So, the energy functional (12) will be

$$E(\phi) = \int_M g^{ii}(\xi)\, d\xi = \int_0^1 \cdots \int_0^1 \mathrm{Tr}(g^{-1})\sqrt{\det(g)}\, d\xi^1 \cdots d\xi^n. \tag{13}$$

The harmonic mapping in this case gives harmonic coordinates on the surface. According to the Hamilton–Shoen–Yau theorem [16, 17, 32] there exists a unique harmonic map $M \to \mathrm{N}$, which is one-to-one, provided that N has nonpositive curvature and convex boundary. These conditions are obviously satisfied for the unit cube $0 < \mu^i < 1$, $i = 1, ..., n$ in $\mathscr{R}^n$. Hence, nondegenerate harmonic coordinates may be constructed on surfaces.

The direct generalization of the Winslow method to the case of two-dimensional adaptive grids is based on the functional (13) written for a two-dimensional surface with local coordinates ξ, η in three-dimensional space (x, y, z). In this case

$$g_{11} = \mathbf{r}_\xi^2 = x_\xi^2 + y_\xi^2 + z_\xi^2,$$
$$g_{12} = g_{21} = (\mathbf{r}_\xi \cdot \mathbf{r}_\eta) = x_\xi x_\eta + y_\xi y_\eta + z_\xi z_\eta,$$
$$g_{22} = \mathbf{r}_\eta^2 = x_\eta^2 + y_\eta^2 + z_\eta^2,$$
$$g^{11} = g_{22}/\det(g), \quad g^{22} = g_{11}/\det(g),$$
$$g^{12} = g^{21} = -g_{12}/\det(g),$$

where $\mathbf{r}_\xi = (x_\xi, y_\xi, z_\xi)^{\mathrm{T}}$ and $\mathbf{r}_\eta = (x_\eta, y_\eta, z_\eta)^{\mathrm{T}}$.

The functional (13) takes the form

$$I = \int_0^1 \int_0^1 (g^{11} + g^{22})\sqrt{\det(g)}\, d\xi\, d\eta = \int_0^1 \int_0^1 \frac{\mathbf{r}_\xi^2 + \mathbf{r}_\eta^2}{J}\, d\xi\, d\eta, \tag{14}$$

where $J = \sqrt{\det(g)} = [g_{11}g_{22} - g_{12}^2]^{1/2} = [\mathbf{r}_\xi^2 \mathbf{r}_\eta^2 - (\mathbf{r}_\xi \mathbf{r}_\eta)^2]^{1/2}$ is the Jacobian of the mapping.

Let the surface be defined as $z = f(x, y)$, where $f \in C^1$. The expression for the Jacobian will be

$$J = (x_\xi y_\eta - x_\eta y_\xi)(1 + f_x^2 + f_y^2)^{1/2}.$$

Substituting this expression into (14) with $z_\xi = f_x x_\xi + f_y y_\xi$, and $z_\eta = f_x x_\eta + f_y y_\eta$, we obtain the functional from [34]

$$I_a = \int_0^1 \int_0^1 \frac{(x_\xi^2 + x_\eta^2)(1 + f_x^2) + (y_\xi^2 + y_\eta^2)(1 + f_y^2) + 2f_x f_y (x_\xi y_\xi + x_\eta y_\eta)}{(x_\xi y_\eta - x_\eta y_\xi)(1 + f_x^2 + f_y^2)^{1/2}} \, d\xi \, d\eta.$$

Now we again consider the grid $(x, y)_{ij}$, $i = 1, ..., n$; $j = 1, ..., m$ and, to simplify the computational formulas, the parametric rectangle $1 < \xi < n$, $1 < \eta < m$ instead of the unit square $0 < \xi < 1$, $0 < \eta < 1$. The functional I_a is approximated by the function

$$I_a^h = \sum_{i=1}^{n-1} \sum_{j=1}^{m-1} \sum_{k=1}^{4} \frac{1}{4} [F_k]_{i+1/2, j+1/2}, \tag{15a}$$

$$F_k = \frac{D_1[1 + (f_x)_k^2] + D_2[1 + (f_y)_k^2] + 2D_3 (f_x)_k (f_y)_k}{J_k [1 + (f_x)_k^2 + (f_y)_k^2]^{1/2}}, \tag{15b}$$

where

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

Derivatives $(f_x)_k$ and $(f_y)_k$ in the kth cell vertex are equal to the corresponding values of derivatives, evaluated in the grid node i, j

$$(f_x)_{ij} = \frac{(f_{i+1,j} - f_{i-1,j})(y_{i,j+1} - y_{i,j-1}) - (f_{i,j+1} - f_{i,j-1})(y_{i+1,j} - y_{i-1,j})}{(x_{i+1,j} - x_{i-1,j})(y_{i,j+1} - y_{i,j-1}) - (x_{i,j+1} - x_{i,j-1})(y_{i+1,j} - y_{i-1,j})},$$

$$(f_y)_{ij} = -\frac{(f_{i+1,j} - f_{i-1,j})(x_{i,j+1} - x_{i,j-1}) - (f_{i,j+1} - f_{i,j-1})(x_{i+1,j} - x_{i-1,j})}{(x_{i+1,j} - x_{i-1,j})(y_{i,j+1} - y_{i,j-1}) - (x_{i,j+1} - x_{i,j-1})(y_{i+1,j} - y_{i-1,j})}.$$

These formulas must be modified for the boundary nodes. Indices "leaving" the computational domain must be replaced by the nearest boundary indices. For example, if $j = 1$, then $(i, j - 1)$ must be replaced by (i, j).

Function (15) possesses the same property as function (6): $I_a^h(G) \to +\infty$ if $G \to \partial D$ for $G \in D$, where D is the set of convex grids, and ∂D is the boundary of the set. To prove this, we first note that the numerator in (15b) is bounded below with the numerator in (6b). Then the proof from Section 1 can be repeated, being applied to function (15).

As before, Eqs. (8), described in Section 1 are used to minimize the function I_a^h. Quantities $(f_x)_{ij}$ and $(f_y)_{ij}$ are assumed to be parameters and therefore all their derivatives in (8) vanish. Note that if $(f_x)_{ij}$ and $(f_y)_{ij}$ vanish, the function I_a^h reduces to the function I^h (6).

The adaptive grid generation algorithm is formulated as follows:

1. Generate a quasi-uniform harmonic grid for the given domain using the unconstrained minimization algorithm as described in Section 1.

2. Compute the values of the control function at each grid node. The result is f_{ij}.

3. Evaluate derivatives $(f_x)_{ij}$ and $(f_y)_{ij}$ using the above formulas.

4. Make one step in the minimization process for the function I_a^h using Eqs. (8) and compute new values of x_{ij} and y_{ij}.

5. Repeat starting with Step 2 to convergency.

It is important that at each step of the iterative process the grid remains convex.

Note that the computational cost of the method is approximately 10 times the cost of the original Winslow method.

In many applications an adaptive grid must be constructed not only within a domain but also on its boundaries. One can do it sequentially [31]: first on the boundaries using the corresponding form of the functional (13) and then within the domain. Let s be the length of a boundary curve and $f(s)$ be a control function. The functional (13) written for a curve with a local coordinate ξ in two-dimensional space (s, f) takes the form (see also [31])

$$I = \int_0^1 \frac{1}{s_\xi (1 + f_s^2)^{1/2}} \, d\xi.$$

The corresponding Euler–Lagrange equation

$$s_\xi (1 + f_s^2)^{1/2} = \text{const}$$

gives the grid $s(\xi)$ as the projection of the uniform grid along the curve $f(s)$ [1, 35].

This method is illustrated by the following example. The square domain $0 < x < 1$, $0 < y < 1$ is considered. The cubic curve

$$y_0(x) = 25(x - 0.5)(x - 0.75)(x - 0.25) + 0.5$$

determines the form of a layer of high gradients. For a given point x, y the function $f(x, y)$ is calculated as

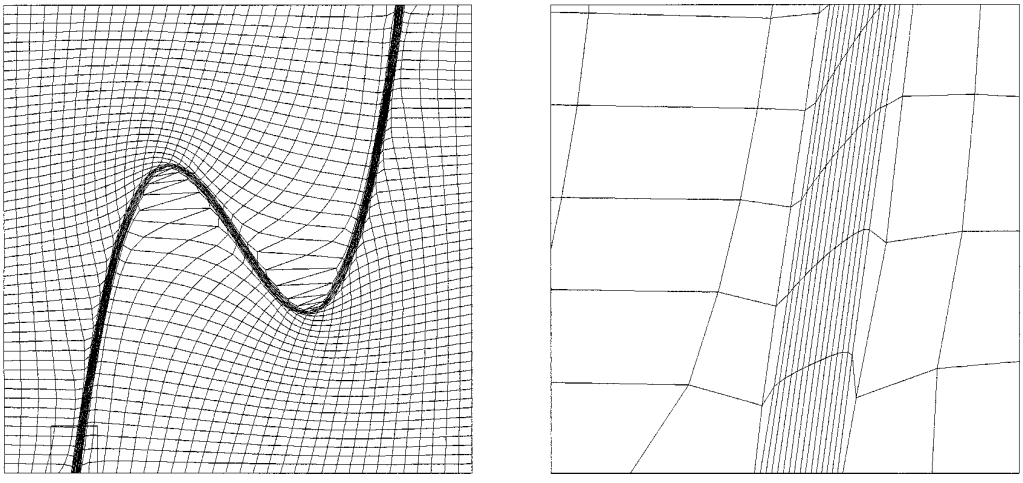

FIG. 10. Adaptive grid, control function (16), the half-width of the layer of high gradients $\delta_0 = 0.01$.

$$f = \begin{cases} 0.5 & \text{if } y \geq y_0 + \delta, \\ 0.25(y - y_0 + \delta)/\delta & \text{if } y_0 + \delta \geq y \geq y_0 - \delta, \\ 0 & \text{if } y \leq y_0 - \delta. \end{cases} \quad (16)$$

Here

$$\delta = \delta_0 \left[1 + \left(\frac{\partial y_0}{\partial x} \right)^2 \right]^{1/2}.$$

The value of δ is chosen so that the width of the layer will be about $2\delta_0$ everywhere along the curve.

Figure 10 shows the 51×51 grid obtained for $\delta_0 = 0.01$. About a third of the grid nodes are within the layer which is very thin and strongly bent. Nevertheless all the grid cells are convex. The enlargement in Fig. 10 shows that the resolution of the layer of high gradients near the boundary is the same as that inside the domain.

However, the grid shown in Fig. 10 has too large a variation in grid spacing and can be unsuitable for obtaining a sufficiently accurate solution of the fluid flow equations. This example is presented here to show the robustness of the algorithm in extreme cases. To construct a grid which is suitable for computations one should use control function (16) multiplied by a coefficient of less than 1. For example, in Fig. 11 the grid obtained with control function (16) multiplied by 0.2 is shown. Variations in grid spacing are not so large, but the number of grid nodes in the layer of high gradients is less (Fig. 10) and so the resolution of the layer is not so high as in the previous example. If in the common case x, y and control function f are scaled to be in the range [0, 1], then the coefficient β before f can be introduced to control the number of grid nodes within the layer of high gradients.

Additional investigations on the adaptive grid generator presented, particularly dealing with the accuracy of the solution of fluid flow equations, will be reported in the near future.

CONCLUSIONS

Details of the two-dimensional structured grid generator have been presented in this paper. The method guarantees that all quadrilateral grid cells are convex at each step of the iterative process. Although other Winslow-type methods are more economical computationally and simpler in implementation, the high reliability of the present method allows us to recommend it for certain problems which require grid lines to be bent strongly in order to obtain a satisfactory grid.

A generalization to the case of adaptive grids based on harmonic maps between surfaces is also considered. First numerical experiments show that the method may be useful for problems with interior curves following a thin layer of high gradients as a control function.

The three-dimensional case is much more complicated than the two-dimensional case, since the simple conditions of Jacobian positiveness cannot be obtained for the trilinear mapping of the unit cube onto a hexahedral cell. The Jacobian of this isoparametric mapping in 3D may be zero or negative even when the eight-corner Jacobians are positive. Faces of hexahedrons are not planar and the notation of convexity also cannot be used. This is why an approach developed for two-dimensional meshes cannot be directly extended to the three-dimensional case. This question is discussed in [37], where one of the possible approaches to the problem is presented and a three-dimensional algorithm with properties similar to the properties of the two-dimensional algorithm presented herein is described.

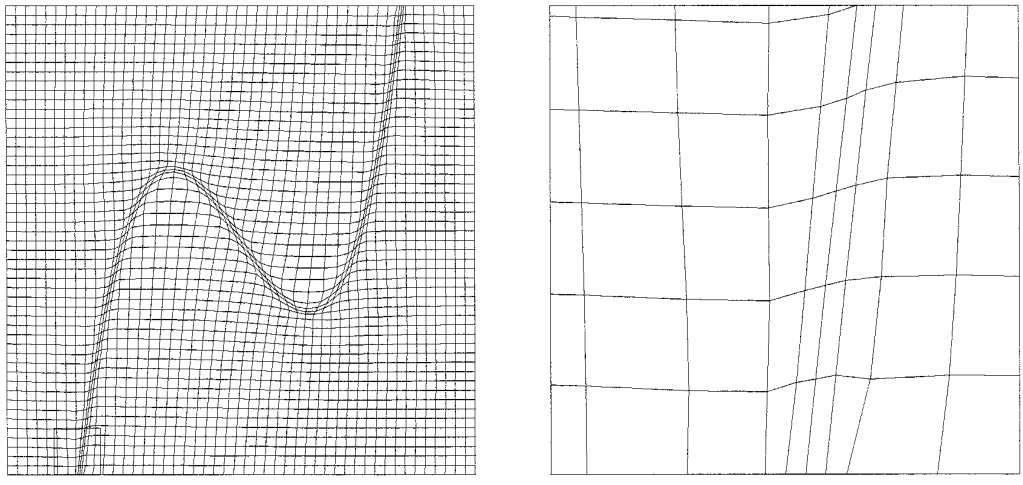

FIG. 11. Adaptive grid, control function (16), multiplied by 0.2.

ACKNOWLEDGMENTS

The authors thank Professor G. P. Prokopov of the Keldysh Institute of Applied Mathematics, Moscow, and Dr. A. G. L. Borthwick of the University of Oxford for helpful discussions related to this work.

REFERENCES

1. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
2. Y. Tu and J. F. Thompson, Three-dimensional solution—Adaptive grid generation on composite configurations, *AIAA J.* **29,** 2025 (1991).
3. *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. Proceedings, Third International Conference, Barcelona, Spain, 3–7 June, 1991*, edited by A. S. Arcilla, J. Hauser, P. R. Eiseman, and J. F. Thompson (North-Holland, New York, 1991).
4. A. M. Winslow, Numerical solution of quasilinear Poisson equation in nonuniform triangle mesh, *J. Comput. Phys.* **1,** 149 (1967).
5. S. K. Godunov and G. P. Prokopov, The use of moving meshes in gas-dynamics calculations, *USSR Comput. Math. Math. Phys.* **12,** No. 2, 182 (1972).
6. J. F. Thompson, F. C. Thames, and C. W. Mastin, Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies, *J. Comput. Phys.* **15,** 299 (1974).
7. P. P. Belinsky, S. K. Godunov, Yu. B. Ivanov, and I. K. Yanenko, The use of a class of quasiconformal mappings to construct difference nets in domains with curvilinear boundaries, *USSR Comput. Math. Math. Phys.* **15,** No. 6, 133 (1975).
8. P. D. Thomas and J. F. Middlecoff, Direct control of the grid point distribution in meshes generated by elliptic equations, *AIAA J.* **18,** 652 (1980).
9. J. U. Brackbill and J. S. Saltzman, Adaptive zoning for singular problems in two dimensions, *J. Comput. Phys.* **46,** 342 (1982).
10. P. J. Roache and S. Steinberg, A new approach to grid generation using a variational formulation, in *Proceedings, AIAA 7th CFD Conference*, edited by U. Chia (Cincinnati, OH, 1985), p. 360.
11. D. A. Anderson, Equidistribution schemes, Poisson generators, and adaptive grids, *Appl. Math. Comput.* **24,** 211 (1987).
12. P. R. Eiseman, Adaptive grid generation, *Comput. Methods Appl. Mech. Eng.* **64,** 321 (1987).
13. J. U. Brackbill, D. B. Kothe, and H. L. Ruppel, FLIP: A low-dissipation, particle-in-cell method for fluid flow, *Comput. Phys. Commun.* **48,** 25 (1988).
14. D. A. Anderson, Grid cell volume control with an adaptive grid generator, *Appl. Math. Comput.* **35,** 209 (1990).
15. Z. U. Warsi and J. F. Thompson, Application of variational methods in the fixed and adaptive grid generation, *Comput. Math. Appl.* **19,** No. 8/9, 31 (1990).
16. A. S. Dvinsky, Adaptive grid generation from harmonic maps on Riemanian manifolds, *J. Comput. Phys.* **95,** 450 (1991).
17. J. U. Brackbill, An adaptive grid with directional control, *J. Comput. Phys.* **108,** 38 (1993).
18. S. R. Kennon and G. S. Dulikravich, Generation of computational grids using optimization, *AIAA J.* **24,** 1069 (1986).
19. S. R. Kennon and G. S. Dulikravich, Composite computational grid generation using optimization, in *Proceedings, Conference on Numerical Grid Generation in Computational Fluid Dynamics*, edited by J. Hauser and C. Taylor (Pinaridge, Swansea, UK, 1986), p. 217.
20. S. A. Ivanenko, Construction of curvilinear grids and their use in finite element method for solving shallow water equations (Computing Center of the USSR Academy of Sciences, Moscow, 1985). [In Russian]
21. S. A. Ivanenko and A. A. Charakhch'yan, An algorithm for constructing curvilinear grids consisting of convex quadrangles, *Soviet Math. Dokl.* **36,** No. 1, 51 (1988).
22. S. A. Ivanenko and A. A. Charakhch'yan, Curvilinear grids of convex quadrilaterals, *USSR Comput. Math. Math. Phys.* **28,** No. 2, 126 (1988).
23. A. A. Charakhch'yan, Almost conservative difference schemes for the equations of gas dynamics, *Comput. Math. Math. Phys.* **33,** 1473 (1993).
24. A. A. Charakhch'yan, Compound difference schemes for time-dependent equations on non-uniform nets, *Commun. Numer. Methods Eng.* **10,** 93 (1994).
25. S. A. Ivanenko, Construction of curvilinear meshes for computing flows in basins, in *Modern Problems in Computational Aerohydrody-*

namics (Mir Publishers, Moscow, and CRC Press, Boca Raton, FL, 1992), p. 211.

26. G. Strang and G. J. Fix, *Analysis of the Finite Element Method* (Prentice–Hall, Englewood Cliffs, NJ, 1973).
27. S. R. Kennon and D. A. Anderson, Unstructured grid adaption for non-convex domains, in *Proceedings, Conference on Numerical Grid Generation in Computational Fluid Mechanics, London, UK, 1988,* edited by S. Sengupta, J. Hansen, P. R. Eiseman, and J. F. Thompson (Pineridge Press, Swansea, UK, 1988), p. 599.
28. S. A. Ivanenko, Generation of non-degenerate meshes, *USSR Comput. Math. Math. Phys.* **28,** No. 5, 141 (1988).
29. S. K. Godunov, A. V. Zabrodin, M. Ya. Ivanov, G. P. Prokopov, and A. N. Kraiko, *Numerical Solution of Multidimensional Problems of Gas Dynamics* (Nauka, Moscow, 1976). [In Russian]
30. I. V. Lomonosov, A. A. Frolova, and A. A. Charakhch'yan, Computation of high-velocity impact of thin foil upon conical target, *Math. Model.*, in press (1997).
31. V. D. Liseikin, Construction of structured grids on n-dimensional surfaces, *USSR Comput. Math. Math. Phys.* **31,** No. 11, 1670 (1991).
32. J. E. Eells and L. Lemaire, Another report on harmonic maps, *Bull. London Math. Soc.* **20,** No. 86, 387 (1988).
33. V. D. Liseikin, On some interpretations of a smoothness functional used in constructing regular and adaptive grids, *Russian J. Numer. Anal. Model.* **8,** 507 (1993).
34. S. A. Ivanenko, Adaptive grids and grids on surfaces, *Comput. Math. Math. Phys.* **33,** 1179 (1993).
35. A. B. White Jr., On selection of equidistributing meshes for two-point boundary-value problems, *SIAM J. Numer. Anal.* **16,** 472 (1979).
36. N. T. Danaev, V. D. Liseikin, and N. N. Yanenko, On numerical calculation of viscous heat-conducting gas flows around curve-related body using curvilinear moving grids, *Chisl. Metody Mekh. Sploshn. Sredy* **11,** No. 1, 51 (1980). [In Russian]
37. S. A. Ivanenko, Harmonic mappings, in *Handbook of Grid Generation* (CRC Press, Boca Raton, FL), in press.