Regularization of the variational method of grid generation

Article in Computational Mathematics and Mathematical Physics · January 1999

2 authors:

Vladimir A. Garanzha
Russian Academy of Sciences
34 PUBLICATIONS 108 CITATIONS

I. E. Kaporin
Russian Academy of Sciences
65 PUBLICATIONS 527 CITATIONS
Regularization of the Barrier Variational Method of Grid Generation

V. A. Garanzha and I. E. Kaporin

Computing Center, Russian Academy of Sciences, ul. Vavilova 40, Moscow, 117967 Russia

Received December 18, 1998; in final form, January 6, 1999

Abstract—A regularization of the barrier variational method for planar grid generation is proposed. The regularization provides a fast and reliable procedure for constructing nondegenerate grids through solution of a discrete minimization problem using only a prescribed cell connectivity structure and given boundary grid points. An efficient numerical method for determining the stationary points of the barrier functional up to the machine truncation error is proposed and tested, which makes it possible to examine the nonuniqueness of stationary points in a number of examples. Numerical experiments have shown that the proposed technique can be used to generate computational grids reliably without any adjustment or control parameters.

1. INTRODUCTION

The key requirements for a grid-generation method include reliability and possibility of implementation without adjusting the method to any specific problem. It is also natural to require that a method be efficient; generate high-quality grids; and ensure the continuity of a grid as depending on input parameters, such as the computational-domain geometry.

Unfortunately, all of these requirements cannot be satisfied simultaneously. As a rule, some compromise has to be made.

Variational methods are rated among the most reliable ones in grid generation in domains of complex geometry. More specifically, we refer to the methods based on quasi-conformal [3, 4] or harmonic [1, 5, 6, 9] mappings. In these approaches, computational grids are generated through minimization of discrete counterparts of the Dirichlet functional and its extensions.

In [7], the Dirichlet functional was used as a basis for constructing a discrete barrier functional that tends to $\infty$ as a nondegenerate grid approaches the boundary of the set of nondegenerate grids.

The principal advantage of the barrier method lies in the fact that there exists at least one stationary point of the discrete functional in the admissible set unless the set is empty. However, the existence of a barrier drastically complicates the procedure of solution of the minimization problem. The main difficulty—the need to pass the barrier from outside—arises when the initial approximation does not belong to the admissible set. Resolution of this difficulty was one of the goals of the study reported here.

In [7], a correction to the Euler–Lagrange equations for the barrier functional was proposed, which makes it possible to reach the interior of the admissible set by using the nonlinear iterative Jacobi method. However, the approach developed in [7] cannot be applied when the initial approximation is too far from the admissible set.

It was found that construction of an admissible approximation is substantially simplified by replacing the original functional with a regularized one whose discrete counterpart is close to the discrete counterpart of the original functional in the admissible set, is a class $C^\infty$ function of its arguments, and tends to $\infty$ away from the admissible set. Under these conditions, implicit minimization methods are highly efficient and reliable and provide solutions irrespective of the choice of an initial approximation.

One important argument in favor of implicit minimization methods is that the computational costs of calculating the functional and its gradient and the solution of local minimization problems are rather high in barrier methods. An optimal alternative is offered by an iterative method in which the fast convergence of a nonlinear iteration is ensured and the computational costs mentioned above are approximately equivalent to those associated with the solution of linear systems required to determine the minimization directions. In
explicit methods, the balance is violated, and the number of nonlinear iteration steps is tens or hundreds times larger as compared to that required in implicit methods while the cost of a single iteration step is lower by a factor of 2 to 4.

One disadvantage of barrier methods is the nonuniqueness of computational grids generated in domains of relatively complex geometry. Obviously, this property is in conflict with the requirement for continuous dependence on input data. A barrier method that ensures the uniqueness of a solution has never been constructed, and conditions under which the existing barrier methods lead to unique solutions have never been formulated. However, numerical experiments show that the grids that correspond to different stationary points within a connected admissible set are rather close to each other. Therefore, the nonuniqueness in barrier methods is less essential as compared to the possible grid degeneration in nonbarrier methods.

The nonuniqueness leads to more stringent requirements for minimization methods. Moreover, one must formulate stopping criteria for minimization that can be satisfied in a reasonable number of iteration steps, which is very important for grid-generation methods designed to work without any adjustment or control parameters.

2. REGULARIZATION OF A HARMONIC FUNCTIONAL

Consider the construction of curvilinear coordinates $\xi(x, y)$ and $\eta(x, y)$ that define a one-to-one mapping of a computational domain $\Omega$ onto a parametric triangle $\mathcal{D}$,

$$\mathcal{D} = \{ \xi_{\min} < \xi < \xi_{\max}, \eta_{\min} < \eta < \eta_{\max} \},$$

through minimization of the Dirichlet functional with metric,

$$\{ \xi(x, y), \eta(x, y) \} = \arg \min_{\Omega} \int \left( \nabla \xi^T \sqrt{g} \nabla \xi + \nabla \eta^T \sqrt{g} \nabla \eta \right) d\sigma$$

(2.1)

assuming that a one-to-one mapping of $\partial \Omega$ onto $\partial \mathcal{D}$ is prescribed. Constraints on the domain $\Omega$ and the class of functions $\xi(x, y)$ and $\eta(x, y)$ required for problem (2.1) to be well posed were given, for example, in [2]. In this paper, the differential problem is formulated as an auxiliary one and serves only as a basis for posing a discrete minimization problem.

In (2.1), $G = \{ g_{ij} \} (i, j = 1, 2)$ is a given metric such that $G = G^T > 0$, and $g = \det G$. Henceforth, all matrix inequalities are interpreted in the sense of matrix definiteness. The metric $G$ can be defined as a function of $\xi$ and $\eta$ (see [3, 4]) or as a function of $x$ and $y$ (see [6, 9]). We consider only the case of $G = G(x, y)$, when (2.1) is a quadratic functional. Construction of the metric is outside the scope of this paper.

Assuming that the Jacobian $J$ of the curvilinear coordinates is positive,

$$J = x_\xi y_\eta - x_\eta y_\xi > 0,$$

(2.2)

one can treat the dependent variables in (2.1) as independent and vice versa to obtain the following minimization problem for the so-called harmonic functional:

$$\{ x(\xi, \eta), y(\xi, \eta) \} = \arg \min_{\Omega} \mathcal{F}, \quad \mathcal{F} = \int_{\mathcal{D}} \frac{g_{11}(x_\xi^2 + y_\xi^2) + g_{22}(y_\eta^2 + y_\eta^2) + 2g_{12}(x_\xi y_\eta + x_\eta y_\xi)}{\sqrt{g}} d\xi d\eta,$$

(2.3)

where the unknown functions

$$x = x(\xi, \eta), \quad y = y(\xi, \eta),$$

map the domain $\mathcal{D}$ onto the computational domain $\Omega$.

In (2.2), (2.3), and further on, the subscripts $\xi$ and $\eta$ denote partial derivatives with respect to $\xi$ and $\eta$, respectively.

Instead of solving minimization problem (2.3) with constraints (2.2), we seek a solution to the following problem: find

$$\{ x(\xi, \eta), y(\xi, \eta) \} = \lim_{\varepsilon \rightarrow 0^+} \arg \min_{x, y} \mathcal{F}_\varepsilon,$$

(2.4)
where
\[
\varphi_\epsilon = \int_\varnothing \frac{g_{11}(x_2^2 + x_3^2) + g_{22}(y_2^2 + y_3^2) + 2g_{12}(x_2y_2 + x_3y_3)}{\chi_\epsilon(J)\sqrt{g}} \, d\xi \, d\eta
\]
and
\[
\chi_\epsilon(J) = \frac{J}{2} + \frac{1}{2} \sqrt{\epsilon^2 + J^2},
\]
and \( \epsilon > 0 \) is sufficiently small. For brevity, we henceforth omit the argument \( J \) of the function \( \chi_\epsilon(J) \). We define \( \chi_\epsilon' = \partial \chi_\epsilon / \partial J \) and \( \chi_\epsilon'' = \partial^2 \chi_\epsilon / \partial J^2 \). The function \( \chi_\epsilon \) has the following properties:
\[
\chi_\epsilon = J \quad \text{as} \quad J \longrightarrow +\infty,
\]
\[
\chi_\epsilon = \frac{1}{4|J|} \epsilon^2 \quad \text{as} \quad J \longrightarrow -\infty,
\] (2.5)
\[
\frac{\chi'\epsilon J}{\chi_\epsilon} \leq 1, \quad 0 \leq \frac{\chi''\epsilon J^2}{\chi_\epsilon} \leq \frac{1}{2} \epsilon^2.
\]
Figure 1 shows the graph of \( \chi_\epsilon(J) \) and its asymptotes \( \epsilon = 0.2 \).

3. APPROXIMATION OF THE FUNCTIONAL AND STATEMENT OF A DISCRETE PROBLEM

Prior to formulating a discrete problem, we introduce the following notation. By \( a_i \) and \( a^j \), we denote the following quantities:
\[
a_1 = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \quad a_2 = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}, \quad a^m \cdot a_k = J \delta_{mk}, \quad a_{mk} = a_m \cdot a_k.
\] (3.1)
Then, we have the relations
\[
a^1 = K_{12}a_2, \quad a^2 = K_{21}a_1, \quad K_{21} = -K_{12} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},
\] (3.2)
and the functional \( \mathcal{J}_\epsilon \) can be rewritten as
\[
\mathcal{J}_\epsilon = \int_\varnothing f_\epsilon(a_1, a_2) \, d\xi \, d\eta, \quad f_\epsilon(a_1, a_2) = \frac{g_{11}a_{11} + g_{22}a_{22} + 2g_{12}a_{12}}{\chi_\epsilon(J)\sqrt{g}}.
\] (3.3)
Let
\[ R = \begin{pmatrix} X \\ Y \end{pmatrix}, \quad X \in \mathbb{R}^{N_x}, \quad Y \in \mathbb{R}^{N_y}, \]
be the vector of x- and y-coordinates of the grid points and \( N \) be the total number of grid points. Suppose that the required computational grid contains \( N \) cells. For a \( c \)th cell, the vector of its vertex coordinates is denoted by
\[ R_c = \begin{pmatrix} X_c \\ Y_c \end{pmatrix}, \quad X_c \in \mathbb{R}^4, \quad Y_c \in \mathbb{R}^4. \]

When a grid cell \( c \) is defined by an ordered set of four integers \( v_1(c), \ldots, v_4(c) \) equal to the numbers of the cell vertices in the complete list of grid points, the following relations are true:
\[ X_c = R_c X, \quad Y_c = R_c Y, \]
where the restriction matrix \( R_c \in \mathbb{R}^{4 \times 4} \) is defined as
\[ R_c = \{ r_{ij} \}, \quad r_{ij} = \begin{cases} 1, & j = v_i(c), \\ 0, & j \neq v_i(c). \end{cases} \]

Suppose that the local vertex numbering in a cell is lexicographic: that is, the vertices of a logical cell defined as a unit square are numbered in the following order: \((0,0), (1,0), (0,1), (1,1)\).

This numbering is chosen to facilitate extension to three-dimensional grids, for which lexicographic numbering is the simplest possible. Without loss of generality, we assume that each logical grid cell is a unit square.

To discretize (3.3), we invoke the finite element method with bilinear basis functions. Since \( x(\xi, \eta) \) and \( y(\xi, \eta) \) are assumed to be bilinear functions of \( \xi \) and \( \eta \) in each cell, the radius vector \( r \) is expressed in terms of the local coordinates as
\[ r(\xi, \eta) = (1-\xi)(1-\eta)r(0,0) + (1-\xi)\eta r(0,1) + \xi(1-\eta)r(1,0) + \xi\eta r(1,1), \]
and the vectors \( \textbf{a}_1 \) and \( \textbf{a}_2 \) are linear functions of \( X_c \) and \( Y_c \) within the cell, respectively. Suppose that the integral in (3.3) is approximately calculated by using some quadrature rule. Then, the value of \( \textbf{a}_{1,2} \) at a quadrature mode \( q(c) \) can be written as
\[ \textbf{a}_1|_{q(c)} = Q_{q(c)} X_c, \quad \textbf{a}_2|_{q(c)} = Q_{q(c)} Y_c, \quad Q_{q(c)} \in \mathbb{R}^{2 \times 4}, \]
(3.4)
where the entries of the matrix \( Q_{q(c)} \) depend on the location of the quadrature point.

Using the notation introduced above, we formulate the discrete counterpart of problem (2.4) as follows.

Find a vector \( \textbf{R} \) that solves the minimization problem
\[ \textbf{R} = \left( \begin{array}{c} X \\ Y \end{array} \right) = \lim_{\varepsilon \to 0, \varepsilon \geq \varepsilon} \inf_{\textbf{R}} F^\varepsilon \left( \begin{array}{c} X \\ Y \end{array} \right) = \sum_{c=1}^{N_c} \sum_{q(c)=1}^{N_q} f^\varepsilon \left( \begin{array}{c} \textbf{a}_1 \\ \textbf{a}_2 \end{array} \right)|_{q(c)} \sigma_{q(c)}, \]
(3.5)
\[ f^\varepsilon \left( \begin{array}{c} \textbf{a}_1 \\ \textbf{a}_2 \end{array} \right) = \frac{\mu_1 a_{11} + \mu_2 a_{22} + 2 \mu_3 a_{12}}{\chi_c(J)\sqrt{\gamma}}, \quad \textbf{a}_{ij} = \textbf{a}_i \cdot \textbf{a}_j, \quad \textbf{a}_1|_{q(c)} = Q_{q(c)} R_c X_c, \quad \textbf{a}_2|_{q(c)} = Q_{q(c)} R_c Y_c, \]
where
\[ X = (I-M)X_x + M X_{int}, \quad Y = (I-M)Y_y + M Y_{int}, \]
(3.6)
and \( M \in \mathbb{R}^{N_x \times N_x} \) is the diagonal matrix (with entries \( m_{ij} \)) in which \( m_{ii} = 1 \) if the \( i \)th grid point is an interior one (i.e., its coordinates are unknown quantities) and \( m_{ii} = 0 \) if the \( i \)th grid point is a boundary one (i.e.,

**COMPUTATIONAL MATHEMATICS AND MATHEMATICAL PHYSICS**

Vol. 39 No. 9 1999
prescribed) point, and

\[ \mathbf{R}_b = \begin{pmatrix} X_b \\ Y_b \end{pmatrix}, \quad \mathbf{R}_{in} = \begin{pmatrix} X_{in} \\ Y_{in} \end{pmatrix}, \]

denote a prescribed vector satisfying the boundary conditions and an unknown vector, respectively.

The weighting factors in the quadrature formula are assumed to satisfy the conditions

\[ \sigma_{q(e)} > 0, \quad \sum_{q(e) = 1}^{N_q} \sigma_{q(e)} = 1. \]

The solution to problem (3.5) can be used as an initial approximation in determining the stationary points of the original barrier functional \( J^h \), where

\[ J^h = \sum_{e = 1}^{N_{el}} \sum_{q(e) = 1}^{N_q} f(a_{1}, a_{2})|_{q(e)} \sigma_{q(e)}, \quad f(a_{1}, a_{2}) = \frac{g_{11}a_{11} + g_{22}a_{22} + 2g_{12}a_{12}}{J \sqrt{g}}, \]

in the admissible set, i.e., in solving the problem

\[ \mathbf{R} = \begin{pmatrix} X \\ Y \end{pmatrix} = \arg \min_{\mathbf{R}} \| \nabla J^h \|_2, \]

where \( \nabla J^h \) is the gradient of functional (3.7), and \( \| \cdot \|_2 \) is the Euclidean norm in \( \mathbb{R}^{2N_e} \).

To construct a barrier approximation, various quadrature formulas can be invoked. In particular, one may use the trapezoid formula to create a barrier at the boundary of the set of convex grids [7]. In this case, \( N_q = 4, \sigma_q = 1/4 \), and the matrices \( Q_1, \ldots, Q_4 \) are

\[
Q_1 = \begin{pmatrix} -1 & -1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \quad Q_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 1 & 1 & 0 \end{pmatrix}.
\]

By using alternative quadrature formulas, a barrier can also be constructed, but the corresponding admissible set consists of grids in which the bilinear mappings have positive Jacobians at the nodes of the chosen quadrature. In particular, barriers are provided by the quadrature formula based on cell-edge midpoints [12] and the four-point Gauss formula. Various examples of barrier approximations can be found in [11, 12].

An analysis of discrete minimization problem (3.5), (3.6) shows that it does not explicitly rely on the fact that the computational grid is structured, even though this was assumed in posing the problem. Thus, discrete functional (3.5), (3.6) can be formally applied for smoothing unstructured grids. To do this, it is sufficient to prescribe the coordinates of grid points and the cell connectivity structure, i.e., the matrices \( R_e \).

4. EXAMPLE OF BEHAVIOR OF THE FUNCTIONAL IN THE CASE OF A GRID WITH A SINGLE INTERIOR POINT

The properties listed in (2.5) are such that \( J^h \) may be expected to tend to \(+\infty\) away from the admissible set so that the rate of its growth is faster for smaller \( \varepsilon \). This proposition cannot be proved rigorously, primarily because no constructive method for measuring the distance to the admissible set is available. Numerical experiments corroborate the hypothesis only in the following empirical sense: the cruder an initial approximation, the larger the value of the functional for a given \( \varepsilon \). This is illustrated by the behavior of the four-cell computational grid shown in Fig. 2.
In this example, the boundary grid points are prescribed, and only the central point, with coordinates $x_0$ and $y_0$, can be optimized. Thus,

$$J^b = J^b(x_0, y_0), \quad J^ε = J^ε(x_0, y_0).$$

The graph of $J^b(x_0, y_0)$ is shown in Fig. 3a, and the graph of $J^ε(x_0, y_0)$ for $ε = 0.001$ is shown in Fig. 3b. Both functions have been scaled with respect to the $z$-axis.

The function $J^b(x_0, y_0)$ has singularities when $J = 0$ at one of the cell vertices. For this reason, the function is plotted in Fig. 3a at distances not less than $δ = 0.02$ from singular lines. It is readily verified that the singular lines are the straight lines shown in Fig. 4.

Figure 3 suggests that $J^ε(x_0, y_0)$ is a convex function on $\mathbb{R}^2$, which can be verified by straightforward calculation. In the general case, this is not true, but it is remarkable that the regularization performed in this example eliminates singularities without leaving any trace, i.e., the smoothed function is not "corrugated."
5. PROPERTIES OF THE FUNCTIONAL

Let us write out the gradient and Hessian of functional (3.5) explicitly, denoting by \( \partial f/\partial a \) the gradient of a function \( f \) with respect to \( a \) (i.e., a column) and by \( \partial f/\partial a^\top \) the transpose of the gradient vector (i.e., a row). We have

\[
\nabla g_e^h = \sum_{c=1}^{N_e} \sum_{q(c)=1}^{N_q} \sigma_{q(c)} \left( \frac{\partial a_1 \partial f_e}{\partial R \partial a_1} + \frac{\partial a_2 \partial f_e}{\partial R \partial a_2} + \frac{\partial a_3 \partial f_e}{\partial R \partial a_3} \right) = \sum_{c=1}^{N_e} \sum_{q(c)=1}^{N_q} \sigma_{q(c)} \begin{pmatrix} M R^e Q^e_{q(c)} & 0 \\ 0 & M R^e Q^e_{q(c)} \end{pmatrix} \begin{pmatrix} \frac{\partial f_e}{\partial a_1} \\ \frac{\partial f_e}{\partial a_2} \end{pmatrix}
\]

and

\[
H = \begin{pmatrix} \frac{\partial^2 g_e^h}{\partial X^\top \partial X} & \frac{\partial^2 g_e^h}{\partial X^\top \partial Y} \\ \frac{\partial^2 g_e^h}{\partial Y^\top \partial X} & \frac{\partial^2 g_e^h}{\partial Y^\top \partial Y} \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}, \quad H_{ij} \in \mathbb{R}^{N_e \times N_q},
\]

\[
H = \begin{pmatrix} I - M & 0 \\ 0 & I - M \end{pmatrix} + \sum_{c=1}^{N_e} \sum_{q(c)=1}^{N_q} \sigma_{q(c)} \begin{pmatrix} M R^e Q^e_{q(c)} & 0 \\ 0 & M R^e Q^e_{q(c)} \end{pmatrix} A_e, \quad A_e = \begin{pmatrix} \frac{\partial^2 f_e}{\partial a_1 \partial a_1} & \frac{\partial^2 f_e}{\partial a_1 \partial a_2} \\ \frac{\partial^2 f_e}{\partial a_2 \partial a_1} & \frac{\partial^2 f_e}{\partial a_2 \partial a_2} \end{pmatrix}
\]

where

\[
A_e = \begin{pmatrix} \frac{\partial^2 f_e}{\partial a_1 \partial a_1} & \frac{\partial^2 f_e}{\partial a_1 \partial a_2} \\ \frac{\partial^2 f_e}{\partial a_2 \partial a_1} & \frac{\partial^2 f_e}{\partial a_2 \partial a_2} \end{pmatrix},
\]

and the expression \( I - M \) in the definition of the Hessian stands for the complement to the total dimension given by an identity matrix.

The matrix

\[
\mathcal{Q} = I - M + \sum_{c=1}^{N_e} \sum_{q(c)=1}^{N_q} \sigma_{q(c)} M R^e Q^e_{q(c)} Q_{q(c)} R_e M
\]

is positive definite for \( Q_{q(c)} \) defined by (3.9) if the diagonal of \( M \) contains at least one zero. Indeed, for each cell, it holds that

\[
\mathcal{Q}_c = \sum_{q(c)=1}^{N_q} Q^e_{q(c)} Q_{q(c)} = \begin{pmatrix} 4 & -2 & -2 & 0 \\ -2 & 4 & 0 & -2 \\ -2 & 0 & 4 & -2 \\ 0 & -2 & -2 & 4 \end{pmatrix} \geq 0,
\]

which implies that, for each cell, the kernel of the matrix \( \mathcal{Q}_c \) is a vector with equal components. Since the kernel vector of \( \mathcal{Q} \) must vanish at boundary grid points, and the computational grid is connected, all these components are zeros. Thus, \( \mathcal{Q} \geq 0 \). This is a well-known fact in the theory of the finite element method [10].

To prove that \( H \) is a positive definite matrix, it would be sufficient to show that

\[
A_e \geq c I, \quad c > 0,
\]

however, this inequality cannot be proved. It is not true even for \( \varepsilon = 0 \) and \( J > 0 \). However, the incorrectness of (5.1) does not imply that the Euler–Lagrange equations for functional (2.3) are of indefinite type when
is a positive definite matrix when \( e = 0 \) and \( J > 0 \), which implies that the principal part of the Euler–Lagrange equations for (2.3) is elliptic. Symmetrization of the off-diagonal blocks of the matrix \( A_0 \) is equivalent to discarding the terms that do not affect the type of the equations.

To write out explicitly the expressions for the derivatives required to calculate the gradient and Hessian, we define

\[
\varphi = g_{11}a_{11} + g_{22}a_{22} + 2g_{12}a_{12}, \quad \alpha = \chi^\epsilon J'/\chi^\epsilon, \quad |\alpha| \leq 1.
\]

Henceforth, we use the pair of indices \( (i, j) \) to denote the cyclic permutation of \( (1, 2) \). In this notation,

\[
\frac{\partial f_\epsilon}{\partial a_i} = \frac{1}{\chi^\epsilon} \left( \frac{2g_{ii}a_i + 2g_{ij}a_j}{\sqrt{g}} - f_\epsilon \chi^\epsilon \alpha_i \right),
\]

\[
\frac{\partial^2 f_\epsilon}{\partial a_i \partial a_i} = \frac{1}{\chi^\epsilon} \left( \frac{2g_{ii}a_i + 2g_{ij}a_j}{\sqrt{g}} \right) - \chi^\epsilon \alpha_i \frac{\partial f_\epsilon}{\partial a_i} - f_\epsilon \chi^\epsilon \alpha_i \alpha_i
\]

\[
\frac{\partial^2 f_\epsilon}{\partial a_i \partial a_j} = \frac{1}{\chi^\epsilon} \left( \frac{2g_{ij}a_i + 2g_{ij}a_j}{\sqrt{g}} \right) - \chi^\epsilon \alpha_i \frac{\partial f_\epsilon}{\partial a_i} - f_\epsilon \chi^\epsilon \alpha_i \alpha_j + f_\epsilon \chi^\epsilon K_{ij},
\]

where \( K_{ij} \) are the matrices defined in (3.2). The properties of diagonal blocks of the matrix \( A_\epsilon \) are of particular interest. They can be written as

\[
\frac{\partial^2 f_\epsilon}{\partial a_i \partial a_i} = P_{ii} - \frac{1}{\chi^\epsilon} f_\epsilon \chi^\epsilon \alpha_i \alpha_i, \quad \text{where} \quad P_{ii} = P_{ii}^0 > 0.
\]

Indeed, the determinants of the matrices \( P_{11} \) and \( P_{22} \) can be expressed as

\[
\det P_{11} = \frac{4}{\chi^\epsilon} \left[ (1 - \alpha)^2 \frac{g_{11}^2}{g} + \alpha^2 \frac{g_{22}^2}{J} \right], \quad \det P_{22} = \frac{4}{\chi^\epsilon} \left[ (1 - \alpha)^2 \frac{g_{11}^2}{g} + \alpha^2 \frac{g_{22}^2}{J} \right],
\]

while their traces have the form

\[
\text{tr} P_{ii} = \frac{1}{\chi^\epsilon} \left[ \frac{4g_{ii}}{\sqrt{g}} - 2\chi^\epsilon \frac{\partial f_\epsilon}{\partial a_i} \right] = \frac{2}{\chi^\epsilon} \left[ \frac{g_{ii}(1 - \alpha) + \alpha^2 \frac{\partial a_i}{J}}{J^2} \right].
\]

Since the invariants of the symmetric 2-by-2 matrices \( P_{11} \) and \( P_{22} \) are positive, the matrices are positive definite.

6. MINIMIZATION METHOD

In Section 3, we formulated discrete minimization problems in closed form. To solve them, we propose the following algorithm.

Choose an initial approximation \( R^0 \).

Set \( R^0 = \text{SCALE} \cdot R^0, \epsilon_0 = \gamma(\epsilon_0, R^0) \).

\textbf{Step 1.} For \( k = 0, 1, 2, \ldots \), find an approximate solution to the problem

\[
\min_{\delta R^k} \mathcal{J}_\epsilon^k(R^k + \delta R^k); \quad (6.1).
\]

set \( R^{k+1} = R^k + \delta R^k, \epsilon_{k+1} = \gamma(\epsilon_k, R^{k+1}) \); if the minimization stopping criterion is satisfied, go to Step 2.

COMPUTATIONAL MATHEMATICS AND MATHEMATICAL PHYSICS Vol. 39 No. 9 1999
Step 2. Set $\varepsilon = 0$; for $m = k + 1, k + 2, \ldots$, find an approximate solution to the problem

$$\min_{\delta R^m} \| \nabla J^h(R^m + \delta R^m) \|; \quad (6.2)$$

set $R^{m+1} = R^m + \delta R^m$; if the minimization stopping criterion is satisfied, go to Step 3.

Step 3. The solution is calculated as $R = \frac{1}{\text{SCALE}} R^{m+1}$.

The function $\gamma$ is defined as

$$\gamma(\varepsilon, R) = \sqrt{\varepsilon^2 + 0.04 \left( \min \left( J_{\text{min}}(R), 0 \right) \right)^2}, \quad (6.3)$$

where $J_{\text{min}}(R)$ is the minimum of the Jacobian $J$ over all grid cells and quadrature points. The parameter $\varepsilon_0$ in (6.1) is a sufficiently small constant; in practical computations, we used $\varepsilon_0 = 10^{-4} - 10^{-9}$. The scaling parameter $\text{SCALE}$ was chosen so that the area of the rectangle that contains all points of the initial grid equals 80 after the scaling has been performed. The choice of these constants and the function $\gamma$ was based on empirical considerations and trials, even though it is important that the method has a convergence speed that weakly depends on control parameters.

Let us consider minimization problem (6.1) with a prescribed $\varepsilon = \varepsilon_1$ in more detail. When the method that determines the minimization direction $P^k$ is prescribed, the calculation of the correction $\delta R^k = \tau_k P^k$ reduces to the minimization of a function of a single variable:

$$\tau_k = \arg \min_{\tau} \left( J_{\text{min}}(R^k + \tau P^k) \right), \quad P^k = -\widetilde{H}^{-1} \nabla J^h_{\varepsilon_k} \quad (6.4)$$

Problem (6.4) was solved by an exhaustive search through the set of $\tau \in \{4, 3, 2, 1, 2^{-1}, \ldots, 2^{-N_\tau}\}$, where $N_\tau$ was either 1 or 2. As a termination criterion for Step 1, we used the condition

$$J_{\text{min}}(R^{k+1}) > 0, \quad \frac{J_{\text{min}}(R^{k+1}) - J_{\text{min}}(R^k)}{\tau_k J_{\text{min}}(R^{k+1})} < \delta_j \quad \text{or} \quad \frac{|J^h_{\varepsilon_k}(R^{k+1}) - J^h_{\varepsilon_k}(R^k)|}{\tau_k J_{\text{min}}(R^{k+1})} < \delta_j, \quad l = 0, 1. \quad (6.5)$$

The value of $\delta_j$ was set equal to 0.01, and $\delta_j$ was usually set equal to 0.0001. The conditions above mean that the relaxation of both $J_{\text{min}}$ and the functional are to be verified. If either condition was fulfilled, then Step 1 was considered as successfully completed.

The matrix $\widetilde{H} = \widetilde{H}(R^k)$ used to calculate the minimization direction $P^k$ was obtained by dropping those terms of the exact Hessian $H$ at the point $R^k$ that make its sign indefinite. The matrix $\widetilde{H}$ was taken in the form

$$\widetilde{H} = \begin{pmatrix} \tilde{H}_{11} & 0 \\ 0 & \tilde{H}_{22} \end{pmatrix}, \quad \tilde{H}_{ii} = I - M + \sum_{c=1}^{N_1} \sum_{q=1}^{N_2} \sigma_{q(c)} M R^c Q^c Q^c P_{i1} \sigma_{q(c)} R.M. \quad (6.6)$$

As shown in the preceding section, $P_{11}$ and $P_{22}$ are positive definite matrices at all quadrature points of every grid cell, which ensures that $\tilde{H}$ is a positive definite matrix. In the general case, it is not necessary to drop so many terms in $\tilde{H}$ to ensure that $H$ is positive definite. However, the key advantage in choosing (6.6) is that the determination of the minimization directions can be reduced to two independent problems of lower dimension. When the computational grid is structured, the matrices $\tilde{H}_{ii}$ in (6.6) can be interpreted as finite-element approximations of the scalar differential operators

$$L_{ii} = -\frac{\partial}{\partial \xi} p_{i1} \frac{\partial}{\partial \eta} p_{i2} - \frac{\partial}{\partial \xi} p_{i2} \frac{\partial}{\partial \eta} p_{i1} - \frac{\partial}{\partial \eta} p_{i1} \frac{\partial}{\partial \xi} p_{i2} - \frac{\partial}{\partial \eta} p_{i2} \frac{\partial}{\partial \xi} p_{i1},$$

where $p_{i1}$, $p_{i2}$ are entries of $P_{ik}$. The matrix $\tilde{H}$ can be further simplified by setting $p_{12} = p_{21} = 0$. Then, $\tilde{H}$ becomes an $M$-matrix, whereby the requirements for the quality of preconditioning are substantially relaxed at the stage of solution of linear systems. However, this simplification is feasible only when the functional is approximated by the trapezoid formula. When other quadrature formulas are applied, the structure of
matrices is not simplified by dropping mixed derivatives. Since we sought various approximations for the functional, we did not consider this simplification.

Minimization problem (6.2) was also reduced to the minimization of a function of a single variable as follows:

\[ \tau_m = \arg \min \| \nabla J^b(R^n + \tau P^n) \|, \quad P^n = -H^{-1} \nabla J^b(R^n). \]  

(6.7)

This problem was solved by an exhaustive search through the set of \( \tau \in \{ 1, 2^{-1}, \ldots, 2^{-N_t} \} \), where \( N_t \) was varied from 4 to 6.

Changing from minimization of the functional to minimization of a residual of the Euler–Lagrange equation is very important for convergence, because the functional is not convex and minimization problem (6.4) may have a solution \( \tau = 0 \) in the neighborhood of a saddle point, in which case the iterative procedure stagnates.

To prove the convergence of the proposed method rigorously, an additional analysis is required. Nonetheless, it can be shown that the following proposition concerning Step 1 is true.

**Proposition.** Suppose that, for any \( k \geq 0 \), the conditions

\[ \varepsilon_{k+1} \leq \varepsilon_k \quad \text{and} \quad J^b_{\varepsilon_k}(R^{k+1}) \leq J^b_{\varepsilon_k}(R^k) + c(G^k)^T \delta R^k, \]  

(6.8)

are satisfied, where \( c \) is a positive constant, and \( G^k \) denotes the vector \( \nabla J^b_{\varepsilon_k}(R^k) \). Then, \( R^k \) converges to a stationary point of the functional \( J^b_{\varepsilon_k} \) at Step 1, where \( \varepsilon_k \) is the limit of the sequence of \( \varepsilon_k \) (\( k = 0, 1, 2, \ldots \)).

To prove this, we use the simple inequality

\[ \varepsilon_{k+1}^2 J^b_{\varepsilon_{k+1}}(R) \leq \varepsilon_k^2 J^b_{\varepsilon_k}(R), \]  

(6.9)

which follows from the definition of the functional and the fact that \( 1/\varepsilon_k = 2(\sqrt{J^2 + \varepsilon_k^2} - J)/\varepsilon_k^2 \). Combining (6.8) with (6.9), we obtain

\[ \varepsilon_{k+1}^2 J^b_{\varepsilon_{k+1}}(R^{k+1}) \leq \varepsilon_k^2 [J^b_{\varepsilon_k}(R^k) - c(G^k)^T \delta R^k]. \]  

(6.10)

Note that

\[ (-G^k)^T \delta R^k = \varepsilon_k (G^k)^T H^{-1} G^k > 0. \]  

(6.11)

by the definition in (6.4). Summing the inequalities in (6.10) over \( k = 0, 1, \ldots, K - 1 \), we obtain

\[ \varepsilon_K^2 J^b_{\varepsilon_K}(R^K) \leq \varepsilon_0^2 J^b_{\varepsilon_0}(R^0) - \sum_{k=0}^{K-1} \varepsilon_k^2 (-G^k)^T \delta R^k. \]  

(6.12)

Since the values of \( \varepsilon_k \) are bounded from below by \( \varepsilon_0 \), inequality (6.12) implies that \( J^b_{\varepsilon_k}(R^k) \) is bounded from above by a constant and

\[ \lim_{k \to \infty} [(-G^k)^T \delta R^k] = 0. \]

Now, to show that the gradient of the functional tends to zero in a reasonable norm, it is sufficient to obtain an upper bound for \( \varepsilon \) that is uniform with respect to \( k \). The largest eigenvalue of \( \varepsilon \) can be estimated by applying the inequality \( P_{\varepsilon} \leq \text{Tr} P_{\varepsilon} \) and Gerschgorin’s lemma, which leads to the following result:

\[ \lambda_{\max}(\varepsilon) \leq 2d \max_{i=1, \ldots, q(c)} \text{Tr} P_{\varepsilon[i_d(c)]}, \]

where \( d \) is the highest valence of a grid point. If the solution vector \( R^k \) is bounded at each iteration step, then it can be readily shown that \( \text{Tr} P_{\varepsilon} \) is uniformly bounded from above. Suppose that the sequence of \( R^k \) is not bounded, that is, contains a subsequence of vectors whose Euclidean norm tends to infinity. Since all grids in this subsequence satisfy prescribed boundary conditions, they must contain a cell with a ratio of sides
that tends to infinity, which obviously contradicts the fact that the functional $\mathcal{F}^{k} (\mathbf{R})$ is uniformly bounded. Therefore, we have the uniform estimate

$$\| \tilde{H} \|_2 \leq c_1,$$

by virtue of (6.11), this entails

$$\frac{1}{c_1} \| G^{k}_1 \|_2 \leq ( - G^{k}_1 )^T \delta \mathbf{R}^k,$$

which proves the proposition.

Thus, to substantiate the iterative method, it is sufficient to show that the practical algorithm satisfies the inequalities in (6.8). This is not quite true in the general case, because the value of $\varepsilon_k$ may increase in the course of iteration. The proof may admit extension to the case when the monotonicity of the sequence of $\varepsilon_k$ is violated a countable number of times. The monotonicity condition can also be imposed by brute force, say, by choosing $\varepsilon_k$ as prescribed by the formula

$$\varepsilon^2_k = \varepsilon^2_k + \varepsilon^3_k \theta^k, \quad k > 0, \quad 0 < \theta < 1.$$ 

If a correct law of decrease in $\varepsilon_k$ is guessed, then the convergence of the iteration at Step 1 is sufficiently fast and, sometimes, even faster than that achieved by using (6.3). On the other hand, if the law of decrease is chosen incorrectly, then $\varepsilon_k$ may drop to a very small value while the solution is still far from the admissible set. As a result, the problem becomes very stiff in many cases, and practical convergence is not achieved because no self-correction mechanism of the kind inherent in (6.3) is provided.

The second inequality in (6.8) follows from the theory of gradient methods (see [17]), because the smoothness of $\mathcal{F}^k$ combined with the fact that $\varepsilon_k \geq \varepsilon_0$ implies that there exists a positive constant $c$ at each iteration step. The key problem here is a uniform lower bound for $c$, which is the subject of our further studies.

7. NUMERICAL EXPERIMENTS

In the numerical experiments discussed here, we considered domains of relatively complex geometry for most of which Winslow’s method (see [8]) was found to lead to degenerate computational grids. Accordingly, any direct comparison between the proposed algorithm and Winslow’s method would be inconsistent. However, it is worth noting that the nonlinear iterations described by (6.1) and (6.2) were rapidly convergent in those cases when Winslow’s method was applicable. All computations were performed on a PC with a 200 Hz Pentium MMX processor. The results reported here were obtained with the metric $G = I$.

To solve linear systems with symmetric positive definite matrices $H_k$, we used the conjugate-gradient method with a preconditioning based on the second-order incomplete Cholesky factorization [13]. One important characteristic of this method is that a symmetric positive definite matrix always admits a stable incomplete triangular factorization of this type. The method does not require diagonal dominance in $H_k$ or $H_k$ being M-matrices. A similar class of stable incomplete triangular factorizations was proposed in [14], but a direct comparison has shown that the incomplete factorization method described in [13] is substantially more efficient than that proposed in [14], provided that comparable memory resources are employed.

Solution of linear systems with the matrix $H$ is a much more difficult task, because $H$ is not a definite matrix. In this approach, the preconditioning was based on a version of the method described in [13], developed for nonsymmetric matrices, and the solution was constructed by a GMRES algorithm [15]. This approach is very reliable, but not the most efficient as applied to the class of problems considered here, because it does not make use of matrix symmetry. Various approaches can be invoked to improve the efficiency of this technique, but lie beyond the scope of this paper.

Figure 5 shows a computational grid constructed to discretize the longitudinal cross section of the interior space of a car. The lower right frame shows a fragment of a computational grid constructed by Winslow’s method. It is clear that the grid is degenerate. If Winslow’s solution is used as an initial approximation in the barrier method, then Step 1 requires one iteration step to reach the admissible set, six additional iteration steps to satisfy criterion (6.5), and five iteration steps of Newton’s method (6.2) to solve the problem up to the machine truncation error. However, if all values at the interior grid points are set to zero as an initial approximation, then the admissible set is reached in 12 iteration steps, seven additional iteration steps are
sufficient to satisfy criterion (6.5), and five iteration steps of Newton’s method (6.2) solve the problem up to the machine truncation error. It took 12.8 s to generate a grid while Step 1 took only 8.4 s to be executed. A fragment of a computational grid corrected by applying the barrier method is shown in the upper right frame in Fig. 5.

Figure 6 illustrates a practical application of the grid shown in Fig. 5. The grid was generated to discretize the interior surface of a car by combining the elliptic barrier generator with an algebraic generator. The grid shown in Fig. 7 is merely the boundary condition for a generator of three-dimensional computational grids.

According to numerical experiments, the choice of $\varepsilon_b$ may affect the speed at which the barrier is passed. For the sake of reliability, it is advisable to prescribe $\varepsilon_b \ll J_{\text{min}}(R)$, where $J_{\text{min}}(R)$ is the expected minimal value of the Jacobian for the scaled computational grid. Otherwise, a stationary point of $J^h$ that lies outside the admissible set may be obtained. The variant with $\varepsilon_b = 10^{-9}$ was found to be effective in all computations, but sometimes the run-time was reduced more than by half as $\varepsilon_b$ was increased to $10^{-4}$. However, all run-times presented here correspond to $\varepsilon_b = 10^{-9}$.

8. NONUNIQUENESS OF THE SOLUTION AND A PRACTICAL CRITERION FOR TERMINATION OF ITERATIONS

Since $J^h$ is a nonconvex functional, it may have several stationary points. Numerical experiments show that this is often the case when the domain has a complex geometry. As a rule, the grids corresponding to different stationary points of the functional within a connected admissible set are almost equivalent in terms of their quality, even though the nonuniqueness of a computational grid, due to the fact that a symmetric solution is a saddle point while asymmetric solutions are local maxima, is a very unfavorable effect. In some special cases, a unique solution can be obtained by changing the metric $G$, but we are unaware of any general solution of the problem.

The technique used to pass the barrier, i.e., Step 1 as defined by (6.1), is characterized by a relatively fast speed even in very complex situations, but the performance of Newton’s method, i.e., Step 2 as defined by (6.2), turns out to be rather erratic in the sense that convergence up to the machine truncation error may be preceded by a long interval of “random walks.” Since the run-times required for
value of the Jacobian $J_{\min}$. However, "jumps" from one stationary point to another are sometimes observed when there exist multiple stationary points. The corresponding minimum of the Jacobian oscillates, sometimes never reaching a steady value. This is the case with any criterion based on an estimated norm of the difference between consecutive approximations. It was noted above that the most reliable criterion, in which the norm of the residual of the Euler–Lagrange equation tends to zero, is frequently too expensive and cannot be cost-effective because all quasi-solutions are approximately equivalent in terms of quality.

Many numerical experiments performed with various domains on grids of various dimensions, including unstructured grids, suggest that a practical termination criterion can be based on (6.5); that is, one may go from Step 1 (6.1) to Step 2 bypassing Newton's method. In particular, a quasi-solution to the problem described above took 3.8 s to be obtained. This effect is much more pronounced when the same problem is solved on a $100 \times 100$ grid. This variant required 53 nonlinear iteration steps to compute a quasi-solution while 41 steps were sufficient to generate a nondegenerate grid. The total run-time was 192 s. Newton's method turned out to be very expensive as implemented in this variant because of a long interval of "random walks" that preceded quadratic convergence. The grid-generation time increased to 929 s without any appreciable gain in quality.

9. CONCLUSIONS

An efficient and reliable method for minimizing the barrier functional has been constructed. The method can be used to generate nondegenerate grids with prescribed cell connectivity structure and prescribed boundary grid points. An efficient implementation of Newton's method that determines the stationary points of the Euler–Lagrange equations made it possible to examine numerically the nonuniqueness of the problem of grid generation by the barrier method.

ACKNOWLEDGMENTS

We thank S.A. Ivanenko for fruitful discussions.

REFERENCES


COMPUTATIONAL MATHEMATICS AND MATHEMATICAL PHYSICS Vol. 39 No. 9 1999


