JOURNAL OF COMPUTATIONAL PHYSICS 74, 422–439 (1988)

Directional Control in Grid Generation

ANTONIOS E. GIANNAKOPOULOS

Department of Engineering, Brown University,
Providence, Rhode Island 02912

AND

ANTONIE J. ENGEL

Department of Applied Mathematics, Brown University,
Providence, Rhode Island 02912

Received June 27, 1986; revised April, 1987

A grid generator in two dimensions with directional control is presented. This generator minimizes an appropriate functional over a finite domain when two reference fields are given. Some problems of uniqueness of the associated partial differential equations were investigated. The directional control of given grids was shown in some simple examples. The directional control should, in general, be combined with other grid generators for mesh adaptation for the solution of actual physical problems. © 1988 Academic Press, Inc.

INTRODUCTION

A functional used for mesh generation $\mathbf{X} = (x(\xi, \eta), y(\xi, \eta))$ can be formulated to optimize a variety of properties [1], including aspect ratio, orthogonality, and grid spacing. Here, we introduce two new functionals that control the orientation of the ξ and η coordinate lines according to given vector fields. The Euler–Lagrange equations that are derived are the partial differential equations whose solution gives a grid oriented according to the specified vector fields.

The vector fields (one for every coordinate line) can be induced from the physics of the problem and are functions of the physical domain. Therefore, the grid generator proposed in this work can be potentially useful for problems that involve mesh adaptation for the solution of problems in magnetodynamics, plasticity, as well as for fluid and solid mechanics that involve anisotropies. It can be shown that the directions of the coordinate lines can be constructed independently, thus accounting for possible anisotropies of the vector fields.

0021-9991/88 $3.00

1. Mathematical Formulation

In the present work we examine a functional in two dimensions, of the form:

$$I_d = \iint_D dx\,dy\,J[(\mathbf{A}\times\nabla\xi)^2 + (\mathbf{B}\times\nabla\eta)^2], \tag{1}$$

where **A** and **B** are directional vector fields given as functions of the physical coordinates x and y, defined on a simply connected finite domain D. These vector fields will be denoted as

$$\mathbf{A} = \begin{pmatrix} A_1(x, y) \\ A_2(x, y) \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} B_1(x, y) \\ B_2(x, y) \end{pmatrix} \tag{2}$$

with A_1, A_2, B_1, and B_2 sufficiently smooth and bounded throughout D, and $\mathbf{A} \neq 0$, $\mathbf{B} \neq 0$ and $\mathbf{A} \neq \mathbf{B}$. In Eq. (1), ξ and η are the coordinates in the logical (computational) domain (Fig. 1a) and J is the Jacobian of the transformation

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{3}$$

with subscripts denoting partial derivatives.

Mesh generation is achieved by minimizing the functional I_d. This has the effect of including the normals to the curves of constant natural coordinates to coalign

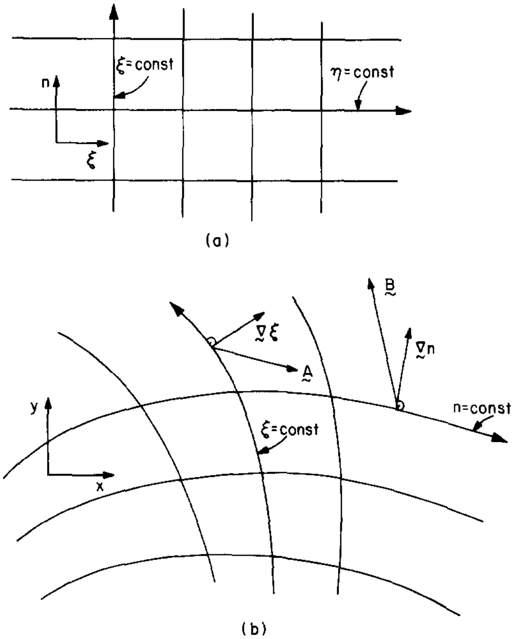

FIG. 1. The directional control in a grid generation: (a) logical domain; (b) physical domain.

with the applied vector fields in the interior of D. This can be seen intuitively by realizing that $\nabla\xi$, for example, is the vector normal to the curve $\xi = \text{const}$ (Fig. 1b). The cross product $[\mathbf{A} \times \mathbf{B}]$ is proportional to the area of the parallelogram formed by $\mathbf{A}$ and $\nabla\xi$. Hence, this area is minimized when these two vectors are colinear. Since the functional is positive definite we are guaranteed a minimum, which in fact is zero for an infinite domain.

The solution of this variational problem is tractable by means of the Euler–Lagrange equations for zero variations of the boundary [2], given by

$$(\partial/\partial x - \partial^2/\partial\xi\, \partial x_\xi - \partial^2/\partial\eta\, \partial x_\eta)\, F = 0 \tag{4}$$

and

$$(\partial/\partial y - \partial^2/\partial\xi\, \partial y_\xi - \partial^2/\partial\eta\, \partial y_\eta)\, F = 0. \tag{5}$$

In our problem, with the appropriate interchange of variables, the functional given by (1) becomes

$$I_d = \int_1^M \int_1^N F\, d\xi\, d\eta, \tag{6}$$

where F is the kernel of the functional, given by

$$F = G^2 + H^2 \tag{7}$$

with

$$G = A_1 x_\eta + A_2 y_\eta = \mathbf{A} \cdot \mathbf{X}_\eta \tag{8}$$

and

$$H = B_1 x_\xi + B_2 y_\xi = \mathbf{B} \cdot \mathbf{X}_\xi. \tag{9}$$

A full derivation of Eqs. (7), (8), and (9) is presented in Appendix A.1. The limits of Eq. (6) imply that the discretization of the computational domain accounts for integer values of ξ and η. It can be further observed that (1) tends to make the grid lines parallel to the vector field, whereas (6) tends to make the grid lines perpendicular to the vector field. This indicates the difference in the formulation of the minimization problem in the physical and the computational space respectively.

From Eqs. (4), (5), and (7) follow

$$\begin{aligned}
&[G(x_\eta(\partial A_1/\partial x) + y_\eta(\partial A_2/\partial x)) + H(x_\xi(\partial B_1/\partial x) + y_\xi(\partial B_2/\partial x))] \\
&\quad -[H(\partial B_1/\partial\xi) + G(\partial A_1/\partial\eta) + B_1(x_\xi(\partial B_1/\partial\xi) + y_\xi(\partial B_2/\partial\xi)) \\
&\quad + A_1(x_\eta(\partial A_1/\partial\eta) + y_\eta(\partial A_2/\partial\eta))] \\
&= x_{\xi\xi} B_1^2 + x_{\eta\eta} A_1^2 + y_{\xi\xi} B_1 B_2 + y_{\eta\eta} A_1 A_2
\end{aligned} \tag{10}$$

$$\begin{aligned}
&[G(x_\eta(\partial A_1/\partial y) + y_\eta(\partial A_2/\partial y)) + H(x_\xi(\partial B_1/\partial y) + y_\xi(\partial B_2/\partial y))] \\
&\quad -[H(\partial B_2/\partial \xi) + G(\partial A_2/\partial \eta) + B_2(x_\xi(\partial B_1/\partial \xi) + y_\xi(\partial B_2/\partial \xi)) \\
&\quad +A_2(x_\eta(\partial A_1/\partial \eta) + y_\eta(\partial A_2/\partial \eta))] \\
&= x_{\xi\xi} B_1 B_2 + x_{\eta\eta} A_1 A_2 + y_{\xi\xi} B_2^2 + y_{\eta\eta} A_2^2 \qquad (11)
\end{aligned}$$

The intermediate results for the various terms that appear in Eqs. (10) and (11) are given in Appendix A.2. Equations (10) and (11) can be solved for x and y, giving the physical coordinates of the directional controlled grid. Notice that these equations have no cross second-order derivatives.

In the degenerated case where $A_1 B_2 = A_2 B_1$, Eqs. (10) and (11) reduce to one equation, and the right-hand side of both (10) and (11) becomes

$$x_{\xi\xi} B_1^2 + x_{\eta\eta} A_1^2 + y_{\xi\xi} k B_1^2 + y_{\eta\eta} k A_1^2 \qquad (12a)$$

for $k = A_2/A_1 = B_2/B_1$; A_1 or $B_1 \neq 0$, and

$$x_{\xi\xi} k B_2^2 + x_{\eta\eta} k A_2^2 + y_{\xi\xi} B_2^2 + y_{\eta\eta} A_2^2 \qquad (12b)$$

for $k = A_1/A_2 = B_1/B_2$; A_2 or $B_2 \neq 0$.

In this case the fields **A** and **B** are parallel. This leads to a degenerate mesh where ξ and η lines are parallel. Therefore, a natural constraint for the vector fields is that their components should satisfy $A_1 B_2 \neq A_2 B_1$ in D.

In addition to the functional given by Eq. (1), an alternative functional appropriate for directional control can be stated as

$$I_d = \iint_D dx\, dy [(\mathbf{A} \times \nabla \xi)^2 + (\mathbf{B} \times \nabla \eta)^2]. \qquad (13)$$

Note that the functional given by (13) is dimensionless (scale independent), whereas the functional given by (1) is not. Following the previous method we can obtain the set of Euler–Lagrange equations for (13) given by

$$\begin{aligned}
&J^2 \left\{ G\left(\frac{\partial A_1}{\partial x} x_\eta + \frac{\partial A_2}{\partial x} y_\eta \right) + H\left(\frac{\partial B_1}{\partial x} x_\xi + \frac{\partial B_2}{\partial x} y_\xi \right) - \left(\frac{\partial B_1}{\partial \xi} H + \frac{\partial A_1}{\partial \eta} G \right) \right\} \\
&\quad -(J^2 B_1 - y_\eta HJ)\left(x_\xi \frac{\partial B_1}{\partial \xi} + y_\xi \frac{\partial B_2}{\partial \xi} \right) + (-y_\xi HJ)\left(x_\xi \frac{\partial B_1}{\partial \eta} + y_\xi \frac{\partial B_2}{\partial \eta} \right) \\
&\quad +(y_\eta GJ)\left(x_\eta \frac{\partial A_1}{\partial \xi} + y_\eta \frac{\partial A_2}{\partial \xi} \right) - (J^2 A_1 + y_\xi GJ)\left(x_\eta \frac{\partial A_1}{\partial \eta} + y_\eta \frac{\partial A_2}{\partial \eta} \right) \\
&= x_{\xi\xi}\{(J^2 B_1 - y_\eta HJ) B_1 + y_\eta(y_\eta(G^2 + H^2) - B_1 HJ)\} \\
&\quad + x_{\eta\eta}\{(J^2 A_1 + y_\xi GJ) A_1 + y_\xi(y_\xi(G^2 + H^2) + A_1 GJ)\} \\
&\quad + x_{\xi\eta}\{y_\xi HJB_1 - y_\eta GJA_1 - y_\xi(y_\eta(G^2 + H^2) - B_1 HJ) \\
&\quad - y_\eta(y_\xi(G^2 + H^2) + A_1 GJ)\}
\end{aligned}$$

$$
\begin{aligned}
&+y_{\xi\xi}\{(J^2B_1-y_\eta HJ)\,B_2-x_\eta(y_\eta(G^2+H^2)-B_1HJ)\}\\
&+y_{\eta\eta}\{(J^2A_1+y_\xi GJ)\,A_2-x_\xi(y_\xi(G^2+H^2)+A_1GJ)\}\\
&+y_{\xi\eta}\{y_\xi HJB_2-y_\eta GJA_2+x_\xi(y_\eta(G^2+H^2)-B_1HJ)\\
&+x_\eta(y_\xi(G^2+H^2)+A_1GJ)\}
\end{aligned}
\tag{14}
$$

$$
\begin{aligned}
&J^2\left\{G\left(\frac{\partial A_1}{\partial y}x_\eta+\frac{\partial A_2}{\partial y}y_\eta\right)+H\left(\frac{\partial B_1}{\partial y}x_\eta+\frac{\partial B_2}{\partial y}y_\xi\right)-\left(\frac{\partial B_2}{\partial \xi}H+\frac{\partial A_2}{\partial \eta}G\right)\right\}\\
&\quad-(J^2B_2+x_\eta HJ)\left(x_\xi\frac{\partial B_1}{\partial \xi}+y_\xi\frac{\partial B_2}{\partial \xi}\right)+(x_\xi HJ)\left(x_\xi\frac{\partial B_1}{\partial \eta}+y_\xi\frac{\partial B_2}{\partial \eta}\right)\\
&\quad+(-x_\eta GJ)\left(x_\eta\frac{\partial A_1}{\partial \xi}+y_\eta\frac{\partial A_2}{\partial \xi}\right)-(J^2A_2-x_\xi GJ)\left(x_\eta\frac{\partial A_1}{\partial \eta}+y_\eta\frac{\partial A_2}{\partial \eta}\right)\\
&=x_{\xi\xi}\{(J^2B_2+x_\eta HJ)\,B_1+y_\eta(-x_\eta(G^2+H^2)-B_2HJ)\}\\
&\quad+x_{\eta\eta}\{(J^2A_2-x_\xi GJ)\,A_1+y_\xi(-x_\xi(G^2+H^2)+A_2GJ)\}\\
&\quad+x_{\xi\eta}\{-x_\xi HJB_1+x_\eta GJA_1-y_\xi(-x_\eta(G^2+H^2)-B_2HJ)\\
&\quad-y_\eta(-x_\xi(G^2+H^2)+A_2GJ)\}\\
&\quad+y_{\xi\xi}\{(J^2B_2+x_\eta HJ)\,B_2-x_\eta(-x_\eta(G^2+H^2)-B_2HJ)\}\\
&\quad+y_{\eta\eta}\{(J^2A_2-x_\xi GJ)\,A_2-x_\xi(-x_\xi(G^2+H^2)+A_2GJ)\}\\
&\quad+y_{\xi\eta}\{-x_\xi HJB_2-y_\eta GJA_2+x_\xi(-x_\xi(G^2+H^2)-B_2HJ)\\
&\quad+x_\eta(-x_\xi(G^2+H^2)+A_2GJ)\}.
\end{aligned}
\tag{15}
$$

2. Boundary Conditions

Let us suppose that we have one solution of the system of partial differential equations (10) and (11) for an unbounded region. Clearly, lines with $\xi=\text{const}$ and $\eta=\text{const}$ are natural boundaries that impose no further problems. The general boundary problem of a finite simply connected domain D was worked on the following simpler problem.

Assuming constant vector fields, Eqs. (10) and (11) give

$$
\begin{aligned}
0&=x_{\xi\xi}B_1^2+x_{\eta\eta}A_1^2+y_{\xi\xi}B_1B_2+y_{\eta\eta}A_1A_2\\
0&=x_{\xi\xi}B_1B_2+x_{\eta\eta}A_1A_2+y_{\xi\xi}B_2^2+y_{\eta\eta}A_2^2.
\end{aligned}
\tag{16}
$$

Equations (16) form a hyperbolic system. Denote by

$$
\begin{aligned}
X&=A_1x+A_2y\\
Y&=B_1x+B_2y.
\end{aligned}
\tag{17}
$$

Then, since $A_1B_2 - A_2B_1 = \Delta \neq 0$, Eqs. (16) can be written in characteristic form as

$$\begin{aligned} X_{\xi\xi} &= 0 \\ Y_{\eta\eta} &= 0. \end{aligned} \tag{18}$$

The general solution of (18) is

$$\begin{aligned} X &= \xi g_1(\eta) + h_1(\eta) \\ Y &= \eta g_2(\xi) + h_2(\xi). \end{aligned} \tag{19}$$

Equations (19) and (17), give a general solution of the form

$$\begin{aligned} x &= \Delta^{-1}[B_2(\xi g_1(\eta) + h_1(\eta)) - A_2(\eta g_2(\xi) + h_2(\xi))] \\ y &= \Delta^{-1}[A_1(\eta g_2(\xi) + h_2(\xi)) - B_1(\xi g_1(\eta) + h_1(\eta))] \end{aligned} \tag{20}$$

with $g_1(\eta)$, $g_2(\xi)$, $h_1(\eta)$, $h_2(\xi)$ to be determined from appropriate boundary conditions.

At this point it is important to note that the boundaries of the physical domain D may not be solutions of (16) and in fact we will not force them to be, therefore, excluding possible boundary layer formation. We will consider the case where the points at the boundaries may move freely on the curves that define them. Let the boundary of D be constructed by four curves (Fig. 2), denoted as

$$\begin{aligned} &(x(0, \eta(, y(0, \eta)); \eta = 0, N \\ &(x(M, \eta), y(M, \eta)); \eta = 0, N \\ &(x(\xi, 0), y(\xi, 0)); \xi = 0, M \\ &(x(\xi, N), y(\xi, N)); \xi = 0, M, \end{aligned} \tag{21}$$

where the relations between x and y are assumed known for every boundary curve. We would like to investigate the boundary conditions that make this problem well

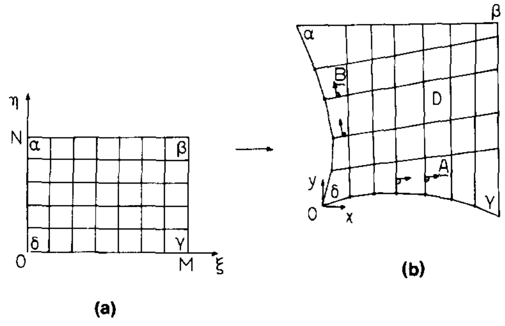

FIG. 2. Initial characteristic problem for **A** and **B** constant directional fields (Riemann's problem).

posed and unique. Notice that the problem stated in the computational space is an initial value problem where the boundaries may serve as initial lines with the solution propagating from these lines.

Assume that the boundaries (21a) and (21c) are described with discrte fixed nodes that are part of the solution (20). We may then write

$$\begin{aligned}
x(0,\eta)&=\Delta^{-1}[B_2h_1(\eta)-A_2(\eta g_2(0)+h_2(0))]\\
y(0,\eta)&=\Delta^{-1}[A_1(\eta g_2(0)+h_2(0))-B_1h_1(\eta)]\\
x(\xi,0)&=\Delta^{-1}[B_2(\xi g_1(0)+h_1(0))-A_2h_2(\xi)]\\
y(\xi,0)&=\Delta^{-1}[A_1h_2(\xi)-B_1(\xi g_1(0)+h_1(0))].
\end{aligned}\tag{22}$$

Inverting (22), we have

$$\begin{aligned}
h_1(\eta)&=A_1x(0,\eta)+A_2\,y(0,\eta)\\
h_2(\xi)&=B_1x(\xi,0)+B_2\,y(\xi,0).
\end{aligned}\tag{23}$$

Taking arbitrarily, $x(0,0)=y(0,0)=0$, we find

$$\begin{aligned}
h_1(0)&=0\\
h_2(0)&=0\\
\eta g_2(0)&=B_1x(0,\eta)+B_2\,y(0,\eta)\\
\xi g_1(0)&=A_1x(\xi,0)+A_2\,y(\xi,0).
\end{aligned}\tag{24}$$

From (20), it can be shown that

$$\begin{aligned}
x_\eta&=\Delta^{-1}[B_2(\xi g_1'(\eta)+h_1'(\eta))-A_2\,g_2(\xi)]\\
x_\xi&=\Delta^{-1}[-A_2(\eta g_2'(\xi)+h_2'(\xi))+B_2\,g_1(\eta)]\\
y_\eta&=\Delta^{-1}[-B_1(\xi g_1'(\eta)+h_1'(\eta))+A_1\,g_2(\xi)]\\
y_\xi&=\Delta^{-1}[A_1(\eta g_2'(\xi)+h_2'(\xi))-B_1\,g_1(\eta)]
\end{aligned}\tag{25}$$

with primes denoting the derivatives with respect to the corresponding argument.

The two additional conditions may indicate with what slope the mesh lines evolve from the initial curves (i.e., from physical considerations):

$$\begin{aligned}
(x_\eta\,y_\eta)\cdot(A_1A_2)^{\mathrm{T}}&=0\\
(x_\xi\,y_\xi)\cdot(B_1B_2)^{\mathrm{T}}&=0.
\end{aligned}\tag{26}$$

Equations (26) with (25) can be integrated

$$\begin{aligned}
g_1(\eta)&=g_1(0)\\
g_2(\xi)&=g_2(0)
\end{aligned}\tag{27}$$

with $g_1(0)$ and $g_2(0)$ given by (24c) and (24d), respectively.

The solution of this problem (initial characteristic problem) gives straight lines of $\xi = \text{const}$ starting from the fixed (21c) boundary, normal to **A** at that boundary and ending on the (21d) boundary. The $\eta = \text{const}$ lines are also straight lines starting from the fixed points of the (21a) boundary, normal to **B** at that boundary and ending on the (21b) boundary (Fig. 2). Obviously, the boundary points on the (21b) and (21d) curves are determined from the solution of the problem. From the above analysis it can be shown that for certain combinations of boundary curves and direction field vectors, pathological cases may arise. Solution curves may intersect the boundary curves more than once or may not intersect them at all, or intersect each other, etc. Therefore, for a computationally acceptable grid, the directional control functions should be used in addition to other mesh generators (i.e., a linear combination with the smoothness functional [1]) that ensure uniqueness when general types of domains, boundary conditions, and direction fields are used. In this paper we will not address these problems, assuming appropriate boundary curves and direction fields.

3. Numerical Solution: Locking of the Mesh

The Euler–Lagrange equations (10), (11) or (14), (15) can be solved by finite difference methods. The standard finite difference approximations for x are given by

$$\begin{aligned}
x_\xi &= (x_{i+1,j} - x_{i-1,j})/2 \\
x_\eta &= (x_{i,j+1} - x_{i,j-1})/2 \\
x_{\xi\xi} &= x_{i+1,j} - 2x_{i,j} + x_{i-1,j} \\
x_{\xi\eta} &= (x_{i+1,j+1} + x_{i-1,j-1} - x_{i+1,j+1} - x_{i-1,j+1})/4 \\
x_{\eta\eta} &= x_{i,j+1} - 2x_{i,j} + x_{i,j-1}
\end{aligned} \tag{28}$$

and similarly for y. In this case index i denotes the position of the ith node along the ξ coordinate and along j for the η coordinate. Note that ξ and η are normalized to take only integer values in the computational space.

The system (10), (11) of second-order partial differential equations with respect to x and y, was numerically integrated by the Jacobi iteration method [1]. The method minimizes the first variation of the residuals,

$$\begin{aligned}
\varepsilon_1 &= B_1 B_2 y_{\xi\xi} + A_1 A_2 y_{\eta\eta} + B_1^2 x_{\xi\xi} + A_1^2 x_{\eta\eta} - s_1 \\
\varepsilon_2 &= B_2^2 y_{\xi\xi} + A_2^2 y_{\eta\eta} + B_1 B_2 x_{\xi\xi} + A_1 A_2 x_{\eta\eta} - s_2,
\end{aligned} \tag{29}$$

where s_1 and s_2 are the source terms given by the left-hand side of (10) and (11), respectively.

This leads to direct vectorization of the increments of the solution for each iteration. The increments, δx_{ij} and $\delta y_{ij}(i, j=\xi, \eta)$, of the solution are given by solving the system:

$$\begin{aligned} &\varepsilon_1 + (\partial\varepsilon_1/\partial x_{ij})\,\delta x_{ij} + (\partial\varepsilon_1/\partial y_{ij})\,\delta y_{ij} = 0 \\ &\varepsilon_2 + (\partial\varepsilon_2/\partial x_{ij})\,\delta x_{ij} + (\partial\varepsilon_2/\partial y_{ij})\,\delta y_{ij} = 0 \end{aligned} \tag{30}$$

(no summation over i and j). The iteration continues untill the increments δx_{ij} and δy_{ij} become less than a specified maximum tolerance. For the functional (1) we have

$$\begin{aligned} \partial\varepsilon_1/\partial x_{ij} &= -2(B_1^2 + A_1^2) \\ \partial\varepsilon_2/\partial x_{ij} &= -2(B_1 B_2 + A_1 A_2) \\ \partial\varepsilon_1/\partial y_{ij} &= -2(B_1 B_2 + A_1 A_2) \\ \partial\varepsilon_2/\partial y_{ij} &= -2(B_2^2 + A_2^2). \end{aligned} \tag{31}$$

The exact forms of the increments are given in Appendix A.3.

By locking of the mesh, we imply that the increments, δx_{ij} and δy_{ij}, are both zero at the first iteration and therefore the points of the initial mesh remain at their original position with $F \neq 0$. This is a feature of the Jacobi iteration (linearization of the original system). Throughout the discussion of the locking phenomenon we will assume that all variables are evaluated at the initial configuration. We will confine our discussion to the system (10) and (11).

From (A.23) in Appendix A, it is obvious that locking may appear when the following conditions are both true

$$\delta x_{ij} = \delta y_{ij} = 0; \qquad i, j = \xi, \eta$$

or, equivalently,

$$\begin{aligned} \varepsilon_1(A_2^2 + B_2^2) &= \varepsilon_2(A_1 A_2 + B_1 B_2) \\ \varepsilon_1(A_1^2 + B_1^2) &= \varepsilon_2(A_1 A_2 + B_1 B_2) \end{aligned} \tag{32}$$

in the whole domain D. From (32) it can be shown that locking occurs, if and only if, the following conditions hold:

(a) $\varepsilon_1 = 0$ and $A_1 A_2 + B_1 B_2 = 0$ or

(b) $\varepsilon_1 = 0$ and $\varepsilon_2 = 0$

everywhere in D, where ε_1 and ε_2 are given by (29).

These locking conditions involve the directional fields, as well as the initial mesh configuration. Suppose, for example, that the initial guess for the mesh involves linear variations of x and y with respect to ξ and η. If in addition **A** and **B** are constant vectors, then $\varepsilon_1 = \varepsilon_2 = 0$ and therefore the initial mesh locks. It is apparent that for such cases the initial mesh should be modified.

This shortcoming of Jacobi iteration method together with the boundary condition analysis, motivated a boundary moving algorithm, such that locking is prevented and at the same time the correct slope of the mesh is assigned at the boundaries. The main steps of the algorithm are summarized as follows:

(1) Before each iteration the boundary points are updated from the interior points adjacent to the boundaries. This can be done by projecting the points adjacent to some part of the boundaries onto the corresponding boundary lines and keeping the other boundary points stationary, so that the problem remains well posed. The updated boundary points are on the initial curves that define them and the grid temporarily has the correct slope at these points.

(2) The problem is solved by the Jacobi method, with the boundary points modified, but fixed, for every iteration step.

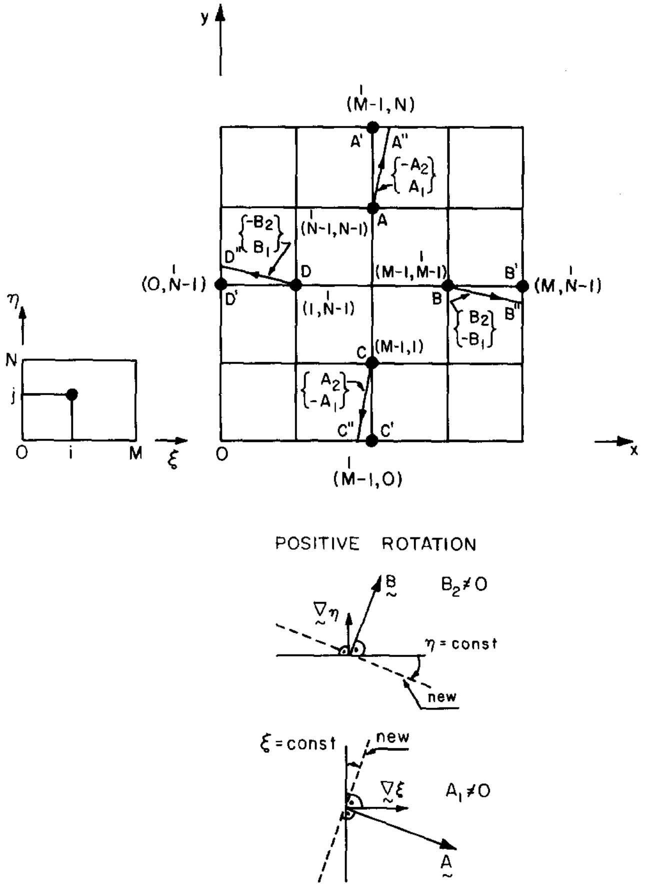

FIG. 3. The algorithm for the boundary point movement.

With this modification, part of the boundary points are moving along the lines that define them and their final position depends on the solution itself. The procedure is shown schematically in Fig. 3 for an orthogonal domain D.

4. Implementation of the Numerical Solution Examples

The directional control functionals (1) and (13) have been implemented in addition to the orthogonal functional, given by [1]

$$I_0 = \iint_D dx\, dy (\nabla\xi \cdot \nabla\eta)^2. \tag{33}$$

The program can generate mesh configurations for simply connected quadrilaterals and annular domains, using Dirichlet boundary conditions for every iteration. Functional (1) or (13) can be used to control the directionality of the mesh.

The vector fields were parametrized as

$$A_1(x, y) = (a_{10} + a_{11}y + a_{12}y^2)(a_{13} + a_{14}x + a_{15}x^2), \tag{34}$$

where $a_{10}, a_{11}, ..., a_{15}$ are constants described by the particular problem to be solved, and similarly for the remaining components A_2, B_1, and B_2. The derivatives of the components of the vector fields were implemented consistnetly with the field parametrization. Therefore, both **A** and **B** and their derivatives with respect to x and y were explicitly included as function subroutines. The program permits using a composite functional, combining both the orthogonal, I_o, and the directional, I_d, control functionals. The coupling is linear

$$I = (1 - \lambda) I_0 + \lambda I_d \tag{35}$$

with parameter $\lambda \in [0, 1]$.

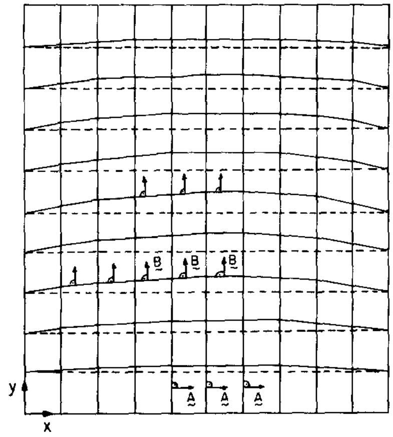

FIG. 4. Final mesh for example 1 (directional control). The initial mesh is shown with dashed lines.

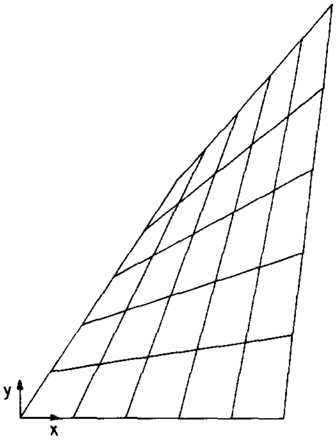

FIG. 5. Initial mesh for example 2.

The boundary movement algorithm was implemented as an option. For such cases we consider only orthogonal, simply connected domains, assuming small rotational movements of the mesh points, in comparison with the minimum initial mesh spacing. This ensures that boundary points will not be propagated off their corresponding boundary lines and is certainly not a fundamental constraint of the problem.

The parametrization of the vector field may be chosen differently than (34); it can be interpolated from values at initial nodal points, or it can be combined with a desirable physical solution of a specific problem (grid adaptation). In addition to (33), other functionals may be combined, to control the smoothness and spacing of the grid.

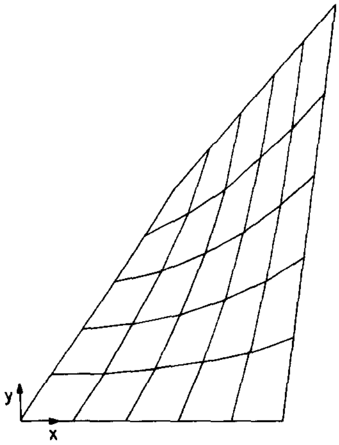

FIG. 6. Final mesh for example 2 (orthogonal control).

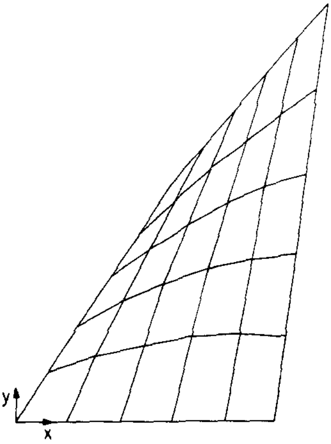

FIG. 7. Final mesh for example 2 (directional control).

The following simple examples were solved numerically. The solution of these examples can be easily derived analytically as well. In all of them, monotonic convergence was observed, with the number of iterations increased for more refined meshes. The vector fields are shown on the final grids obtained by the proposed method.

1. In the first example, we start with an orthogonal, uniform mesh (Fig. 4) and we impose a directional field given by:

$$\mathbf{A}=\begin{pmatrix}1\\0\end{pmatrix},\qquad \mathbf{B}=\begin{pmatrix}(-0.5+0.2x)(0.5y-0.1y^2)\\1\end{pmatrix}. \tag{36}$$

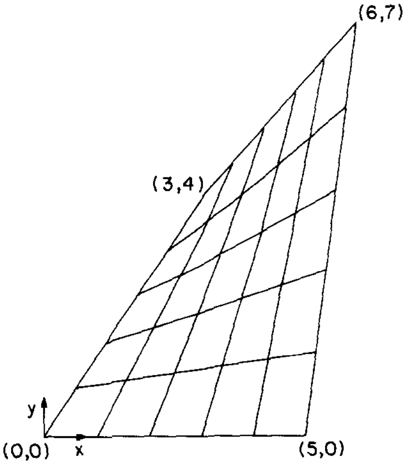

FIG. 8. Final mesh for example 2 (combined directional and orthogonal control, $\lambda = 0.5$).

For this example, the functional (1) was used, with Dirichlet boundary conditions. The final mesh configuration is shown in Fig. 5. Notice that the η-lines remain straight and ξ-lines become arcs, conforming with the imposed directional field. The solution is scale depended due to the form of the functional used. The imposed field satisfies the boundary conditions and locking is prevented due to the nonuniformity of the vector fields.

2. In the second example, we impose the same vector fields given by (36), to a skewed quadrilateral. The final configuration resulting from the orthogonal control only, is shown in Fig. 6. Figure 7 shows the final configuration for the directional control for the same domain. The influence of the combined functional (35) can be seen in Fig. 8, for $\lambda = 0.5$ and for the same initial configuration. Fixed boundary conditions and the functional (1) were used in this case. The orthogonality functional (33), proved to have a pronounced influence over the directional functional (1).

3. The boundary conditions greatly affect the solution. To demonstrate this, we examined the solution of a strip (Fig. 9a) with **A** and **B** set to the constant vectors

$$\mathbf{A} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \qquad \mathbf{B} = \begin{pmatrix} -0.3 \\ 1 \end{pmatrix} \tag{37}$$

that represent rotation of the η-lines. The solution requires that the boundary points move to accommodate the motion of the mesh inside the domain (Fig. 9b). Note that the directional functional can control ξ and η lines independently. The

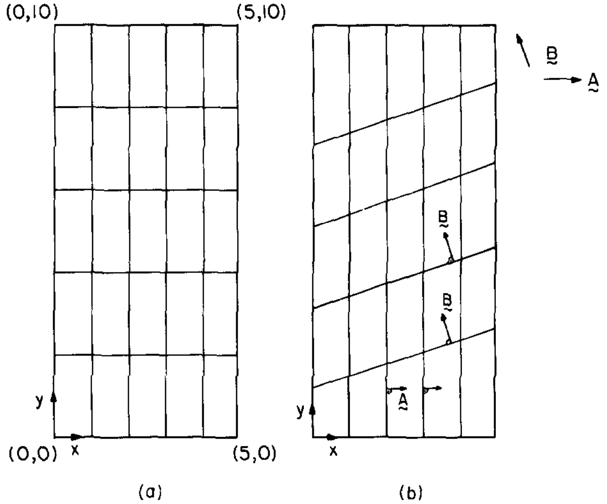

FIG. 9. (a) Initial mesh for example 3; (b) final mesh for example 3 (directional control).

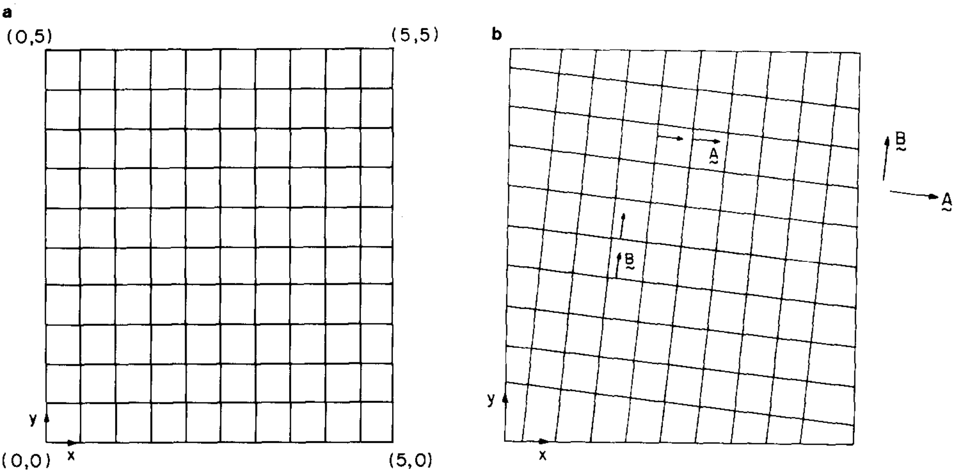

FIG. 10. (a) Initial mesh for example 4; (b) final mesh for example 4 (directional control).

functional (13) was used, in this case, in addition to the moving boundary algorithm.

4. In the fourth example, we used the functional (13) for a rectangular domain (Fig. 10a). The directional field is described by

$$\mathbf{A} = \begin{pmatrix} 1 \\ -0.1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 0.1 \\ 1 \end{pmatrix}, \tag{38}$$

representing a rigid rotation of the mesh. The resulting mesh is given in Fig. 10b. The algorithm for moving the boundary points was used in this case also. It can be observed that the solution is scale independed.

CONCLUSIONS

In this paper we have demonstrated how Eqs. (10) and (11), as well as (14) and (15) can be used to control the direction of a given mesh. The directional control of such a computational mesh is expected to be very useful for solving numerically partial differential equations using adaptive meshing procedures for finite difference or finite element techniques.

Boundary conditions seem to play a very important role in determining the stability and uniqueness of the mesh generated with the directional control. Locking phenomena can be present due to specific combinations of directional field vectors, initial mesh spacing, and boundary conditions. In such cases, we have developed one simple way to imose boundary conditions of Neumann type so that the problem remains well posed. Other control functionals can be included to form a most general combined functional for generating meshes with postulated properties [1].

The present ideas can be easily extended to account for domains of arbitrary shape and connectivity, as well as for other types of boundary conditions and integration algorithms.

Appendix A

1. Derivation of the kernel F of the functional:

$$I_d = \iint F \, d\xi \, d\eta \tag{A.1}$$

$$(\mathbf{A} \times \nabla\xi)^2 = (A_1 \xi_y - A_2 \xi_x)^2 \tag{A.2}$$

$$(\mathbf{B} \times \nabla\eta)^2 = (B_1 \eta_y - B_2 \eta_x)^2. \tag{A.2}$$

Interchanging variables we get

$$\xi_x = y_\eta / J \tag{A.3}$$

$$\xi_y = -x_\eta / J \tag{A.4}$$

$$\eta_x = -y_\xi / J \tag{A.5}$$

$$\eta_y = x_\xi / J \tag{A.6}$$

$$dx \, dy = J \, d\xi \, d\eta. \tag{A.7}$$

Therefore,

$$\iint J\{(\mathbf{A} \times \nabla\xi)^2 + (\mathbf{B} \times \nabla\eta)^2\} \, dx \, dy = \iint \{(A_1 x_\eta + A_2 y_\eta)^2 + (B_1 x_\xi + B_2 y_\xi)^2\} \, d\xi \, d\eta. \tag{A.8}$$

Therefore,

$$F = F(x, y, x_\eta, x_\xi, y_\eta, y_\xi; \xi, \eta). \tag{A.9}$$

2. Derivation of the Euler–Lagrange equations:

$$\frac{\partial F}{\partial x} = 2\left\{G\left(x_\eta \frac{\partial A_1}{\partial x} + y_\eta \frac{\partial A_2}{\partial x}\right) + H\left(x_\xi \frac{\partial B_1}{\partial x} + y_\xi \frac{\partial B_2}{\partial x}\right)\right\} \tag{A.10}$$

$$\frac{\partial F}{\partial y} = 2\left\{G\left(x_\eta \frac{\partial A_1}{\partial y} + y_\eta \frac{\partial A_2}{\partial y}\right) + H\left(x_\xi \frac{\partial B_1}{\partial y} + y_\xi \frac{\partial B_2}{\partial y}\right)\right\} \tag{A.11}$$

$$\frac{\partial F}{\partial x_\xi}=2HB_1, \qquad \frac{\partial F}{\partial y_\xi}=2HB_2 \tag{A.11}$$

$$\frac{\partial F}{\partial x_\xi}=2GA_1, \qquad \frac{\partial F}{\partial y_\eta}=2GA_2 \tag{A.12}$$

$$\frac{\partial}{\partial \xi}\frac{\partial F}{\partial x_\xi}=2\left\{H\frac{\partial B_1}{\partial \xi}+B_1\frac{\partial H}{\partial \xi}\right\} \tag{A.13}$$

$$\frac{\partial}{\partial \eta}\frac{\partial F}{\partial x_\eta}=2\left\{G\frac{\partial A_1}{\partial \eta}+A_1\frac{\partial G}{\partial \eta}\right\} \tag{A.14}$$

$$\frac{\partial}{\partial \xi}\frac{\partial}{\partial y_\xi}=2\left\{H\frac{\partial B_2}{\partial \xi}+B_2\frac{\partial H}{\partial \xi}\right\} \tag{A.15}$$

$$\frac{\partial}{\partial \eta}\frac{\partial}{\partial y_\eta}=2\left\{G\frac{\partial A_2}{\partial \eta}+A_2\frac{\partial G}{\partial \eta}\right\}. \tag{A.16}$$

From the chain rule,

$$\frac{\partial H}{\partial \xi}=x_\xi\frac{\partial B_1}{\partial \xi}+B_1x_{\xi\xi}+y_\xi\frac{\partial B_2}{\partial \xi}+B_2y_{\xi\xi} \tag{A.17}$$

$$\frac{\partial G}{\partial \eta}=x_\eta\frac{\partial A_1}{\partial \eta}+A_1x_{\eta\eta}+y_\eta\frac{\partial A_2}{\partial \eta}+A_2y_{\eta\eta} \tag{A.18}$$

with no summation over *i* and *j*. Also,

$$\frac{\partial A_1}{\partial \xi}=\frac{\partial A_1}{\partial x}x_\xi+\frac{\partial A_1}{\partial y}y_\xi \tag{A.19}$$

$$\frac{\partial A_1}{\partial \eta}=\frac{\partial A_1}{\partial x}x_\eta+\frac{\partial A_1}{\partial y}y_\eta \tag{A.20}$$

and the same for

$$\frac{\partial A_2}{\partial \xi}, \quad \frac{\partial A_2}{\partial \eta}, \quad \frac{\partial B_1}{\partial \xi}, \quad \frac{\partial B_1}{\partial \eta}, \quad \frac{\partial B_2}{\partial \xi}, \quad \frac{\partial B_2}{\partial \eta}.$$

3. Jacobi iteration scheme:

$$\begin{bmatrix} \dfrac{\partial \varepsilon_1}{\partial x_{ij}} & \dfrac{\partial \varepsilon_1}{\partial y_{ij}} \\ \dfrac{\partial \varepsilon_2}{\partial x_{ij}} & \dfrac{\partial \varepsilon_2}{\partial y_{ij}} \end{bmatrix}\begin{Bmatrix} \delta x_{ij} \\ \delta y_{ij} \end{Bmatrix}=\begin{Bmatrix} -\varepsilon_1 \\ -\varepsilon_2 \end{Bmatrix} \tag{A.21}$$

with no summation over i and j. Inverting (A.21) we get

$$D = \det \begin{bmatrix} \dfrac{\partial \varepsilon_1}{\partial x_{ij}} & \dfrac{\partial \varepsilon_1}{\partial y_{ij}} \\ \dfrac{\partial \varepsilon_2}{\partial x_{ij}} & \dfrac{\partial \varepsilon_2}{\partial y_{ij}} \end{bmatrix} = 4(A_1 B_2 - A_2 B_1)^2 \tag{A.22}$$

$$\begin{Bmatrix} \partial x_{ij} \\ \delta y_{ij} \end{Bmatrix} = \frac{1}{D} \begin{Bmatrix} 2\varepsilon_1(A_2^2 + B_2^2) - 2\varepsilon_2(B_1 B_2 + A_1 A_2) \\ -2\varepsilon_1(B_1 B_2 + A_1 A_2) + 2\varepsilon_2(B_1^2 + A_1^2) \end{Bmatrix}, \tag{A.23}$$

where all terms in (A.23) can be evaluated from the previous iteration.

ACKNOWLEDGMENTS

The authors would like to express their deepest gratitude to Professor J. U. brackbill for his valuable suggestions and for revising and commenting on this paper.

REFERENCES

1. J. U. BRACKBILL AND J. C. SALTZMAN, *J. Comput. Phys.* **46**, 342 (1982).
2. M. BECKER, *The Principles and Applications of Variational Methods*, (Monograph No. 27, MIT Press, Cambridge, MA, 1980).
3. J. F. THOMSON, F. G. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **15**, 299 (1974).