Journal of Computational Physics **171,** 805–821 (2001)
doi:10.1006/jcph.2001.6811, available online at http://www.idealibrary.com on IDEAL®

Nearly Orthogonal Two-Dimensional Grid Generation with Aspect Ratio Control

Volkan Akcelik,* Branislav Jaramaz,†,‡ and Omar Ghattas*

**Laboratory for Mechanics, Algorithms, and Computing, Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania; †Center for Orthopaedic Research, UPMC Shadyside Hospital, Pittsburgh, Pennsylvania; and ‡Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania*
E-mail: volkan@andrew.cmu.edu

Received June 6, 2000; revised April 18, 2001

An improved method for nearly orthogonal grid generation is presented in this study. The generating system is based on solution of a system of partial differential equations with finite difference discretization. To prevent grid lines from collapsing onto each other, the grid cell aspect ratio is controlled by functions that limit excessive ratios. Bounding all the aspect ratios is essential for high-quality numerical approximations using such grid-based methods as finite elements, finite differences, or finite volumes. The influence of the number of grid points, type of boundary, and intensity of the grid quality control function and grid properties are investigated. Specification of both boundary point distribution on all sides and moving boundaries is used. The proposed method is applied to various test problems from the literature. This method is shown to provide a good balance between controlling grid orthogonality and cell aspect ratio. © 2001 Academic Press

Key Words: nearly orthogonal grid generation; aspect ratio control.

1. INTRODUCTION

To solve partial differential equations posed on spatial domains, a collection of points called a grid is imposed on the computational domain. Grid quality is important for minimizing computational error. A well-designed grid should be orthogonal. Additionally, for isotropic problems, grid aspect ratios near one are important for good conditioning of the discrete operator, as well as for reducing errors in derivatives of the approximate solution.

Orthogonal grid generation is the subject of many studies [3–21]. One of the well-known and frequently used ways to obtain orthogonal grids in two dimensions is through conformal mapping. Theoretical foundations of elliptic grid generation owe much to the theory of conformal mappings, which preceded it by several decades. In fact, the main impetus for the development of elliptic methods is the lack of versatility in the construction of conformal maps and inability to control the distribution of grid points in the generated grid [1, 3, 8, 11, 12, 16, 20].

Conformal mapping is restricted to having equal scale factors in all directions, i.e. a region of small circles or rectangles remains as circles or rectangles after transformation [1, 8, 20]. Therefore, such techniques may perform well with respect to aspect ratio, but may perform poorly with respect to orthogonality. To improve orthogonality some researchers have prescribed scale factors, i.e. the scale factor is not unit but rather some adjustable constant throughout the domain [4, 5]. However, the use of prescribed scale factors is still too restrictive for a generally applicable transformation technique [7, 8, 17, 18, 20]. Mobley and Stewart [7] have used a simple increasing function to perform a 1-D stretching transformation on the region after conformal mapping. In this method, the scale factors have to be determined.

Ryskin and Leal [8] generalized Mobley and Stewart's idea to nonconstant scale factors by proposing the covariant Laplace equations as a generating system for the grid coordinates. In this method, constraints on the components of the metric tensor of the curvilinear coordinates are used to achieve orthogonality and to control the spacing of coordinate lines. This method has been the subject of many articles [9–16, 20]. The main problem in all these methods is determination of the so-called distortion function f which controls the scale factors.

Albert [9], Allievi and Calisal [10], and Eça [12] used a method in which the distortion function is calculated in the entire domain during nonlinear iterations. The difference between these approaches is the numerical methods—finite difference or finite element—employed by the authors to solve the equation. As stated by Eça [12], for certain geometries and for some boundary point distributions, the orthogonality constraint becomes too restrictive and one of the scale factors tends to zero. This leads to collapse of grid lines and to an unacceptable grid. In this paper we overcome this deficiency. The distortion function is adjusted automatically during the course of numerical solution to reflect the evolving gradient of the grid coordinates. To prevent grid lines from approaching each other, new functions in the form of pseudo-forces are introduced. They help to adjust the scale factors and therefore control aspect ratios, in addition to the orthogonality constraint. Moreover, prescribing values of the distortion function at the boundaries is explored as a useful tool for resolving boundary layers. Both Dirichlet and Neumann–Dirichlet boundary conditions on the generating system are examined.

The grid quality obtained through the new method is compared to the grid quality of other methods. Conformal mapping produces grid aspect ratios near one, but with weaker orthogonality. In addition, this method is not robust. Eça's method is more robust and results in better orthogonality. However, it leads to poorer aspect ratios. The examples demonstrate that the new method is comparable to Eça's with respect to orthogonality, and to conformal mapping with respect to aspect ratio.

2. THE GRID GENERATING SYSTEM

The generating system used by Ryskin and Leal [8] is based on the simple observation that x and y as Cartesian coordinates in the physical space are linear functions of position. Thus, **grad**(x) and **grad**(y) are constant-valued vector fields, and it follows that

$$\nabla^2 x = 0, \quad \nabla^2 y = 0, \tag{1}$$

for the two-dimensional case, where ∇^2 is the covariant Laplace operator.

In the following we will assume summation on repeated indices. Let g_{ij} be the elements of the covariant metric tensor of a multidimensional coordinate system $\xi^1, \xi^2, \ldots, \xi^n$, which define the length of an arc according to the relation

$$ds^2 = g_{ij} d\xi^i d\xi^j. \tag{2}$$

If g is the determinant of this tensor and g_{ij} are elements of the corresponding contravariant tensor, the Laplace operator in curvilinear coordinates can be written as [21]

$$\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} \left(\sqrt{g} g^{ij} \frac{\partial}{\partial \xi^j} \right). \tag{3}$$

The development of an appropriate coordinate system must begin by specifying the metric tensor, and it is this specification that determines the properties of the resulting coordinate system. For example by setting off-diagonal components to zero, the coordinate system becomes orthogonal. There are always m degrees of freedom (where $m = 2$ in 2-D and $m = 3$ in 3-D) in choosing the mapping functions [8].

For an orthogonal curvilinear coordinate system, off-diagonal elements of the metric tensor must be zero. Also, g^{ii} (contravariant components of metric tensor) are equal to $1/g_{ii}$ for orthogonal coordinates. Therefore, Eq. (1) for two dimensional orthogonal curvilinear coordinates ξ and η becomes

$$\frac{\partial}{\partial \xi} \left(f \frac{\partial x}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\frac{1}{f} \frac{\partial x}{\partial \eta} \right) = 0, \tag{4a}$$

$$\frac{\partial}{\partial \xi} \left(f \frac{\partial y}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left(\frac{1}{f} \frac{\partial y}{\partial \eta} \right) = 0. \tag{4b}$$

In the above equations, $f(\mathrm{x}(\xi, \eta), \mathrm{y}(\xi, \eta))$ is the distortion function which is defined as the ratio of scale factor in the η-direction to that in the ξ-direction, i.e.,

$$f = \frac{h_\eta}{h_\xi}, \tag{5}$$

where the scale factors are defined by

$$h_\eta = \sqrt{g_{22}} = \sqrt{\left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2}, \tag{6a}$$

$$h_\xi = \sqrt{g_{11}} = \sqrt{\left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2}. \tag{6b}$$

Equations (4) have been used extensively for grid generation. The main problem with covariant Laplace operators is their highly nonlinear nature. To linearize (4), some approaches prescribe the distortion function a priori, which causes problems. Since the only constraint in Eqs. (4) is on orthogonality, and if the distortion function f is restricted to unity, (4) reduces to conformal mapping. For a 2-D problem, this second restriction is obviously a major limitation on the class of possible mappings. Indeed, the lack of robustness or “stiffness” of the conformal mapping, which makes it ill-suited for our purpose, is due to this unnecessary restriction. In general, prescribing the distortion function a priori leads to robustness problems in the generating system.

The difference between the various available methods is the way in which the distortion function f is obtained. Three types of procedures have been proposed:

Method 1: Calculate f from its definition (5) at the boundaries and obtain its values in the domain by interpolation or by solving a Laplace equation.

Method 2: Specify a class of admissible functions for f that guarantees the existence of a unique solution.

Method 3: Calculate f from its definition (5) in the entire domain.

The first method corresponds to the "weak constraint" method of Ryskin and Leal [8]. Using this method Chikhliwala and Yortsos [13] calculated f at the boundaries and obtained values of f in the interior using the algebraic interpolation suggested by Ryskin and Leal [8]. They found that the success of this method in producing orthogonal grids in this case was primarily attributed to symmetry of the region. Tamamidis and Assanis [18] used Poisson's equation instead of algebraic interpolation in the entire domain, in order to try to control grid spacing.

In the second method, used by Duraiswami and Prosperetti [16], Kang and Leal [14], and Oh and Kang [15], the main problem is to define an admissible function for f. Ascoli *et al.* [11] showed that if f is a special product of the form $f(\xi, \eta) = \Phi(\xi)\Theta(\eta)$, and if h_ξ is specified at one of the boundaries, then an orthogonal mapping does exist between (ξ, η) and (x, y). However, robustness problems can still arise.

The last method was used by Albert [9], Alievi and Calisal [10], and Eça [12]. Albert used finite difference discretization to solve (4). Alievi and Calisal used a Bubnov–Galerkin procedure to solve (4) and found that it is possible to obtain orthogonal meshes using both symmetrical and unsymmetrical domains. They stated that the success of the approach was due to the Bubnov–Galerkin procedure rather than calculating the distortion function f from its definition (5). Eça also calculated the distortion function directly from its definition, but used finite difference discretization to solve (4). Although he obtained good results, the method appears to have some problems. In some geometries and for certain boundary point distributions, the orthogonality constraint in the domain may cause the collapse of several grid lines into one, creating convergence difficulties in the algorithm for solving (4). Locally, these regions are characterized by one of the scale factors tending to zero. In addition to the difficulties in obtaining a solution to the generating system, the poor cell aspect ratios of the resulting grid lead to poor quality approximations and ill-conditioning for many numerical methods for PDE.

To prevent scale factors approaching zero in some part of the domain, Eça's method is improved in this study by introducing functions that control the scale factors in both ξ- and η-directions. The two-dimensional grid generation system then takes the following form:

$$\frac{\partial}{\partial \xi}\left(f \frac{\partial x}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial x}{\partial \eta}\right) + P_x(h_\xi) + Q_x(h_\eta) = 0, \tag{7a}$$

$$\frac{\partial}{\partial \xi}\left(f \frac{\partial y}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial y}{\partial \eta}\right) + P_y(h_\xi) + Q_y(h_\eta) = 0. \tag{7b}$$

Here $\mathbf{P}(h_\xi)$ and $\mathbf{Q}(h_\eta)$ are inhomogeneous source terms that alter the solution (x, y) in such a way as to control favorably the scale factors h_ξ and h_η, and hence the aspect ratio of the resulting grid.

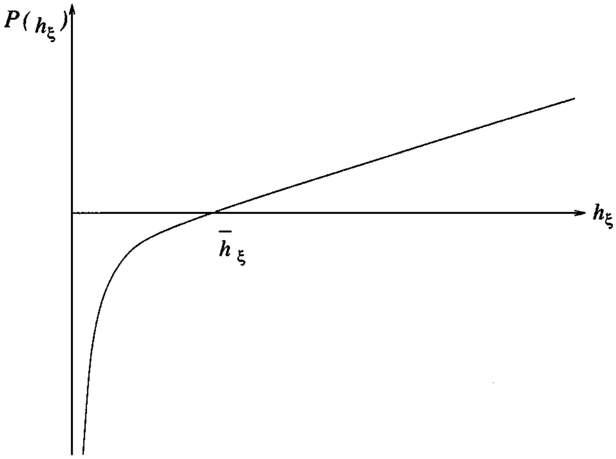

FIG. 1. $\mathbf{P}(h_\xi)$ as a function of h_ξ. Force between grid points becomes repulsive when local scale factor is less than mean scale factor, and attractive when it is greater.

The functions **P** and **Q** are defined by

$$\mathbf{P}(h_\xi) = c\left(h_\xi - \frac{\bar{h}_\xi^2}{h_\xi}\right), \tag{8a}$$

$$\mathbf{Q}(h_\eta) = c\left(h_\eta - \frac{\bar{h}_\eta^2}{h_\eta}\right), \tag{8b}$$

where c is a positive so-called force constant. By changing its magnitude, it is possible to change the intensities of **P** and **Q**. The mean scale factors $\bar{h}_\xi$ and $\bar{h}_\eta$ are defined as

$$\bar{h}_\xi(\eta) = \left.\frac{\int h_\xi \, d\xi}{\int d\xi}\right|_{\eta=const.}, \tag{9a}$$

$$\bar{h}_\eta(\xi) = \left.\frac{\int h_\eta \, d\eta}{\int d\eta}\right|_{\xi=const.}. \tag{9b}$$

The functions **P** and **Q** act like distributed forces proportional to the deviation of the local scale factors from the mean scale factors. As shown in Fig. 1 the pseudo-force is zero when the local scale factor h_ξ is equal to the mean scale factor $\bar{h}_\xi$. It becomes increasingly negative as h_ξ approaches zero, and increasingly positive as h_ξ increases beyond $\bar{h}_\xi$.

3. THE DISCRETIZED GRID GENERATING SYSTEM

The PDEs in Eq. (7) are discretized by the finite difference method. For a typical grid point (Fig. 2), setting $\Delta\xi = 1$ and $\Delta\eta = 1$, we obtain [12]

$$f_{i+1/2,j}\left(\frac{\partial x}{\partial \xi}\right)_{i+1/2,j} - f_{i-1/2,j}\left(\frac{\partial x}{\partial \xi}\right)_{i-1/2,j} + \frac{1}{f_{i,j+1/2}}\left(\frac{\partial x}{\partial \eta}\right)_{i,j+1/2} - \frac{1}{f_{i,j-1/2}}\left(\frac{\partial x}{\partial \eta}\right)_{i,j-1/2} + (P_x)_{i,j} + (Q_x)_{i,j} = 0 \tag{10a}$$

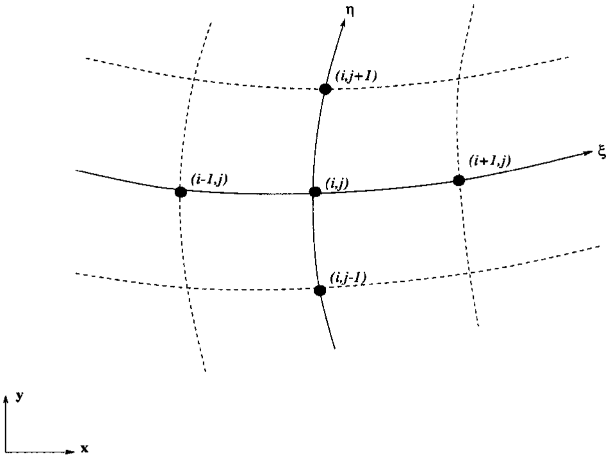

FIG. 2. A typical grid point $(i,\, j)$.

$$f_{i+1/2,j}\left(\frac{\partial y}{\partial \xi}\right)_{i+1/2,j} - f_{i-1/2,j}\left(\frac{\partial y}{\partial \xi}\right)_{i-1/2,j} + \frac{1}{f_{i,j+1/2}}\left(\frac{\partial y}{\partial \eta}\right)_{i,j+1/2}$$
$$- \frac{1}{f_{i,j-1/2}}\left(\frac{\partial y}{\partial \eta}\right)_{i,j-1/2} + (P_y)_{i,j} + (Q_y)_{i,j} = 0. \tag{10b}$$

The indices $(i,\, j)$ refer to the directions ξ and η, respectively. The partial derivative of x with respect to ξ is given by

$$\left(\frac{\partial x}{\partial \xi}\right)_{i,j} = x_{i+1/2,j} - x_{i-1/2,j}, \tag{11}$$

and similar expressions apply for $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$. Expressions for f_{ij} can be found in [12].

Discretized forms of the pseudo-force $\mathbf{P}(\overline{h}_\xi)$ and $\mathbf{Q}(\overline{h}_\eta)$ can be derived as follows. First, $\overline{h}_{\xi_j}$ for a grid line j in the ξ direction, and $\overline{h}_{\eta_i}$ for a grid line i in the η direction (Fig. 2) are defined in discretized form as

$$\overline{h}_{\xi_j} = \frac{1}{M}\sum_{i=1}^{M} h_{\xi_{i,j}}^{i,i+1}, \tag{12a}$$

$$\overline{h}_{\eta_i} = \frac{1}{N}\sum_{j=1}^{N} h_{\xi_{i,j}}^{j,j+1}, \tag{12b}$$

where M and N are number of grid lines in ξ- and η-directions, respectively. Scale factors $h_{\xi_{i,j}}$ and $h_{\eta_{i,j}}$ are discretized as

$$h_{\xi_{i,j}}^{i,i+1} = [(x_{i+1,j} - x_{i,j})^2 + (y_{i+1,j} - y_{i,j})^2]^{\frac{1}{2}}, \tag{13a}$$

$$h_{\eta_{i,j}}^{j,j+1} = [(x_{i,j+1} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2]^{\frac{1}{2}}, \tag{13b}$$

where subscripts $i,\, j$ and superscripts $i, i+1$, and so forth, represent node numbers and discretized directions, respectively. The discretized form of **P** and **Q** corresponding to each line segment will be

$$P_{i,j}^{i,i+1} = c\left[h_{\xi_{i,j}}^{i,i+1} - \overline{h}_{\xi_{i,j}}^{2} \big/ h_{\xi_{i,j}}^{i,i+1}\right], \tag{14a}$$

$$P_{i,j}^{i-1,i} = c\left[h_{\xi_{i,j}}^{i-1,i} - \overline{h}_{\xi_{i,j}}^{2} \big/ h_{\xi_{i,j}}^{i-1,i}\right], \tag{14b}$$

$$Q_{i,j}^{j,j+1} = c\left[h_{\eta_{i,j}}^{j,j+1} - \overline{h}_{\eta_{i,j}}^{2} \big/ h_{\eta_{i,j}}^{j,j+1}\right], \tag{14c}$$

$$Q_{i,j}^{j-1,j} = c\left[h_{\eta_{i,j}}^{j-1,j} - \overline{h}_{\eta_{i,j}}^{2} \big/ h_{\eta_{i,j}}^{j-1,j}\right]. \tag{14d}$$

P and **Q** are defined in ξ- and η-directions. To find their corresponding components in x- and y-directions, the following formulas are used:

$$\begin{aligned}(P_x)_{i,j} + (Q_x)_{i,j} = {} & P_{i,j}^{i,i+1}\left(\frac{\left(\frac{\partial x}{\partial \xi}\right)}{h_\xi}\right)_{i,j}^{i,i+1} + P_{i,j}^{i-1,i}\left(\frac{\left(\frac{\partial x}{\partial \xi}\right)}{h_\xi}\right)_{i,j}^{i-1,i} \\ & + Q_{i,j}^{j,j+1}\left(\frac{\left(\frac{\partial x}{\partial \eta}\right)}{h_\eta}\right)_{i,j}^{j,j+1} + Q_{i,j}^{j-1,j}\left(\frac{\left(\frac{\partial x}{\partial \eta}\right)}{h_\eta}\right)_{i,j}^{j-1,j},\end{aligned} \tag{15a}$$

$$\begin{aligned}(P_y)_{i,j} + (Q_y)_{i,j} = {} & P_{i,j}^{i,i+1}\left(\frac{\left(\frac{\partial y}{\partial \xi}\right)}{h_\xi}\right)_{i,j}^{i,i+1} + P_{i,j}^{i-1,i}\left(\frac{\left(\frac{\partial y}{\partial \xi}\right)}{h_\xi}\right)_{i,j}^{i-1,i} \\ & + Q_{i,j}^{j,j+1}\left(\frac{\left(\frac{\partial y}{\partial \eta}\right)}{h_\eta}\right)_{i,j}^{j,j+1} + Q_{i,j}^{j-1,j}\left(\frac{\left(\frac{\partial y}{\partial \eta}\right)}{h_\eta}\right)_{i,j}^{j-1,j}.\end{aligned} \tag{15b}$$

4. ITERATIVE ALGORITHM

Because the system of equations defined in (7) is nonlinear, a Picard-like iterative algorithm is used to solve it numerically:

1. Choose four corner points of the physical domain that serve as the corner points of the grid in computational domain. Calculate x and y values of the other boundary grid points by dividing physical boundaries into equal segments.
2. Determine an initial approximation for the interior grid points by bilinear interpolation.
3. Calculate the distortion function f from Eqs. (5) and (6).
4. Solve the system of Eqs. (7) with fixed f values calculated in Step 3 using a few iterations of the SOR method. Calculate **P** and **Q** from Eqs. (15).
5. Adjust boundary conditions. If Dirichlet boundaries are applied nothing is done. For sliding (Neumann–Dirichlet) boundaries, relocate boundary nodes to satisfy orthogonality.
6. Go to Step 3, if convergence criteria on orthogonality and aspect ratios are not satisfied.

5. APPLICATION AND COMPARISON

The main purpose of the proposed method is to increase the robustness and flexibility of existing orthogonal grid generation methods. It is preferable that a robust method be one in which the distortion function is not prescribed, but rather determined by the domain. However, using this method without pseudo-forces (**P** and **Q**) may result in the collapse of several grid lines onto each other for some domains and for some boundary point distributions. This may also cause convergence difficulties and deviations from orthogonality [12]. The main impetus for introducing pseudo-forces is to prevent this behavior and to impose some control on grid spacing.

In this section, some characteristics of the new method are explored, and it is compared with existing ones. Several domains commonly used in the literature are selected as grid generation examples. Maximum deviation from orthogonality (MDO), mean deviation from orthogonality (ADO), maximum grid aspect ratio (MAR), and mean grid aspect ratio (AAR) defined below, are used to study quality of the resulting grids.

MDO and ADO are calculated from

$$\mathrm{MDO} = \max_{i,j}(|90^\circ - \theta_{i,j}|), \tag{16a}$$

$$\mathrm{ADO} = \frac{1}{(n_x-2)}\frac{1}{(n_y-2)}\sum_{i=2}^{n_x-1}\sum_{j=2}^{n_y-1}(|90^\circ - \theta_{i,j}|), \tag{16b}$$

where n_x and n_y are the number of grid points in x- and y-directions, respectively, and

$$\theta = \arccos\left(\frac{g_{12}}{h_\xi h_\eta}\right). \tag{17}$$

MAR and AAR are calculated from

$$\mathrm{MAR} = \max_{i,j}\left(\max\left(\frac{h_{\xi_{i,j}}^{i,i+1}}{h_{\eta_{i,j}}^{j,j+1}}, \frac{h_{\eta_{i,j}}^{j,j+1}}{h_{\xi_{i,j}}^{i,i+1}}\right)\right), \tag{18a}$$

$$\mathrm{AAR} = \frac{1}{(n_x-2)}\frac{1}{(n_y-2)}\sum_{i=2}^{n_x-1}\sum_{j=2}^{n_y-1}\max\left(\frac{h_{\xi_{i,j}}^{i,i+1}}{h_{\eta_{i,j}}^{j,j+1}}, \frac{h_{\eta_{i,j}}^{j,j+1}}{h_{\xi_{i,j}}^{i,i+1}}\right). \tag{18b}$$

The optimal values of MDO and ADO are 0°, and for MAR and AAR, 1 (for isotropic PDEs).

Three criteria for convergence are used,

$$\frac{|f^n - f^{n-1}|}{f^n} < 10^{-5}, \tag{19a}$$

$$\max\left(\left|x_{i,j}^n - x_{i,j}^{n-1}\right|, \left|y_{i,j}^n - y_{i,j}^{n-1}\right|\right) < 10^{-6}, \tag{19b}$$

$$|\mathrm{ADO}^n - \mathrm{ADO}^{n-1}| < 10^{-8}, \tag{19c}$$

where superscript n refers to iteration number.

The following are chosen as examples:

1. A circular region (Domain A) is used to study grid characteristics as a function of grid point density (Fig. 3). Different force constants are used.

2. A concave region (Domain B) is used as the second test case (Fig. 4). The effects of the force constant on grid properties are compared. The effects of specifying the distance of the first grid point from the boundary are also studied.

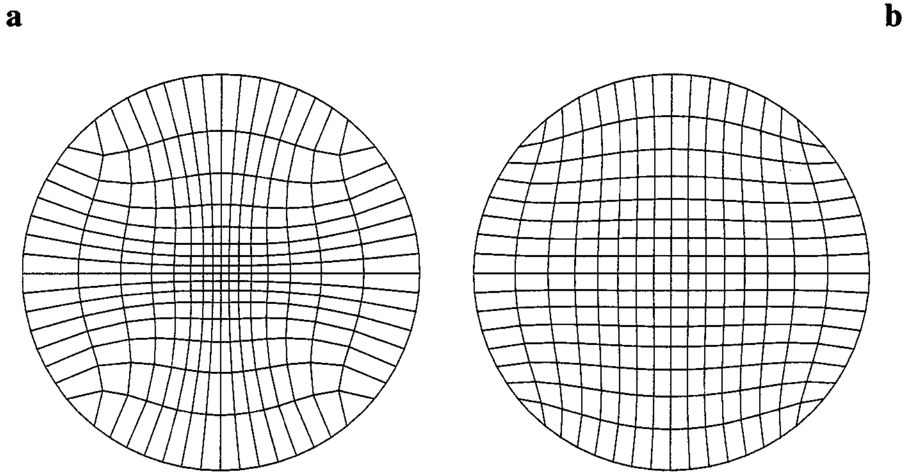

FIG. 3. Domain A. Force constants are 0 and 0.1 in Grids a and b, respectively. Specified boundary conditions on all boundaries. Grid points on all boundaries are equidistant.

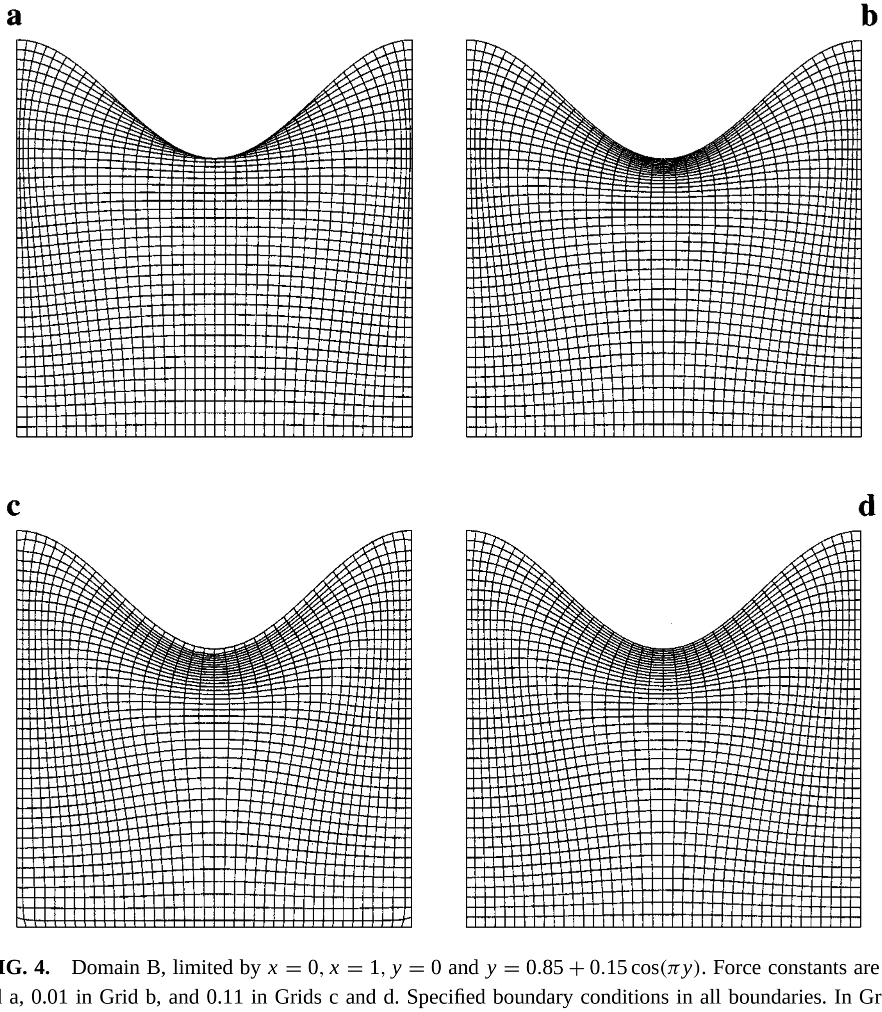

FIG. 4. Domain B, limited by $x = 0, x = 1, y = 0$ and $y = 0.85 + 0.15\cos(\pi y)$. Force constants are 0 in Grid a, 0.01 in Grid b, and 0.11 in Grids c and d. Specified boundary conditions in all boundaries. In Grid c, distances to first grid points are specified on all the boundaries. Grid points on the boundaries $x = 0$, $y = 0$, and $y = 1$ are equidistant; on the top $x = \xi$.

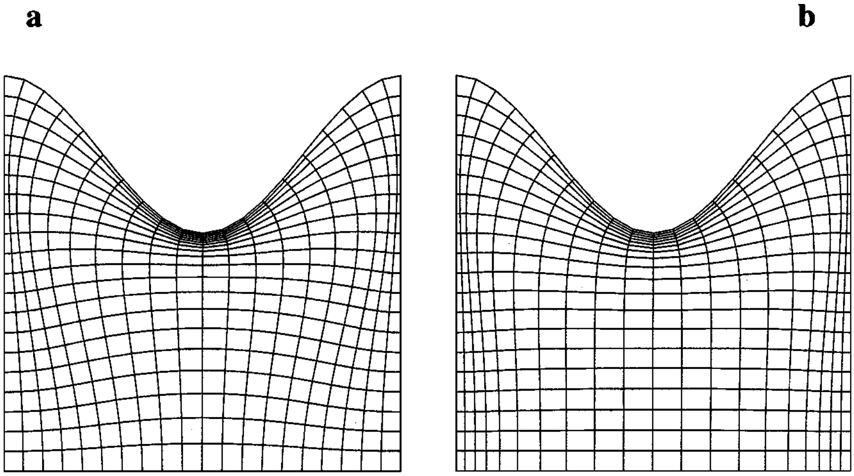

FIG. 5. Domain C limited by lines $x = 0$, $x = 1$, $y = 0$, $y = 0.8 + 0.2\cos(\pi y)$. In Grid a, specified boundaries are used on all boundaries. In Grid b, sliding boundaries are used on the bottom. Distances between grid points are equal along $x = 0$, $y = 0$, and $y = 1$. On the top $x = \xi$. Force constants are 0.01.

3. A concave region (Domain C) similar to domain B but with higher curvature is used as the third test case (Figs. 5 and 6). Both the effects of the force constant and different boundary conditions are examined.

4. A region between two half circles (Domain D) is used as the fourth test case (Figs. 7 and 8). As in the third test case, the effects of using different boundary conditions and force constants are compared.

5. Domain E (Fig. 9) is a region limited by the coordinate axes and lines $y = 1$ and $x = 1/2 + 1/6\cos(\pi y)$. As in the third test case, the effects of using different boundary conditions and force constants are compared.

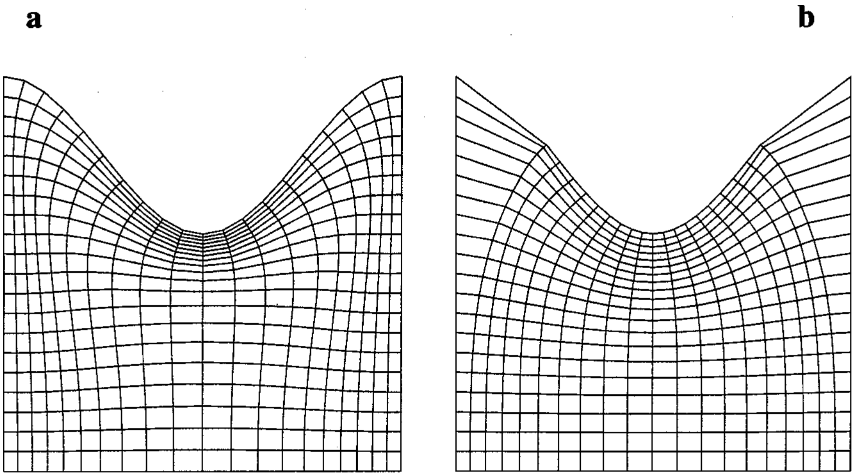

FIG. 6. Domain C. In Grid a, sliding boundaries are used on the bottom; in Grid b sliding boundaries are used both at the bottom and top. Equidistant distribution of grid points on the boundaries $x = 0$, $y = 0$ and $y = 1$. At the top side $x = \xi$. Force constants are 0.11.

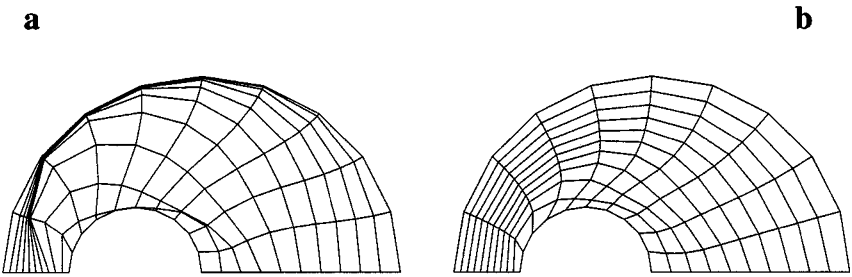

FIG. 7. Domain D, force constants are 0.001 for Grid a and 0.1 for Grid b. Specified boundaries on all sides. Grid points on all boundaries are equidistant.

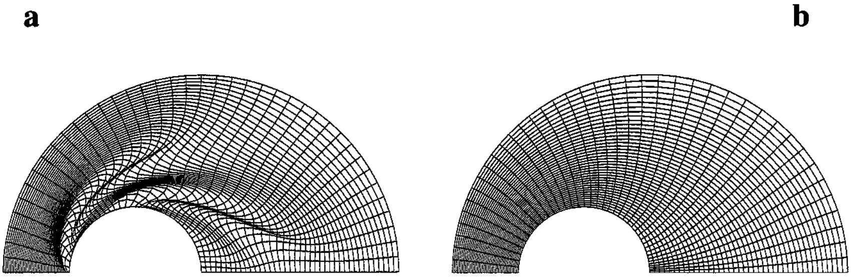

FIG. 8. Domain D, for both grids, force constants are 0.1. In Grid a, specified boundary conditions are used. In Grid b, sliding boundaries are used for circular sides, specified boundaries are used on the other sides. Distances between grid points are equal along all specified boundaries.

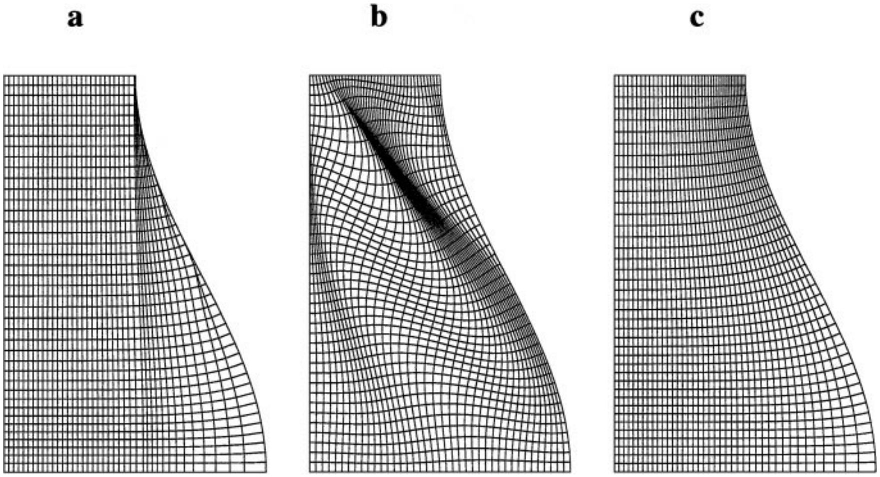

FIG. 9. Domain E, limited by lines $x = 0$, $x = 1$, $y = 0$ and $x = 1/2 + 1/6\cos(\pi y)$. Sliding boundaries are used for the three straight sides in Grid a and c, while specified boundaries are used on the other sides. Force constants are 0 for Grid a, 0.01 for Grid b and c. Along the specified boundaries, grid points are equidistant, while at the left side $y = \xi$.

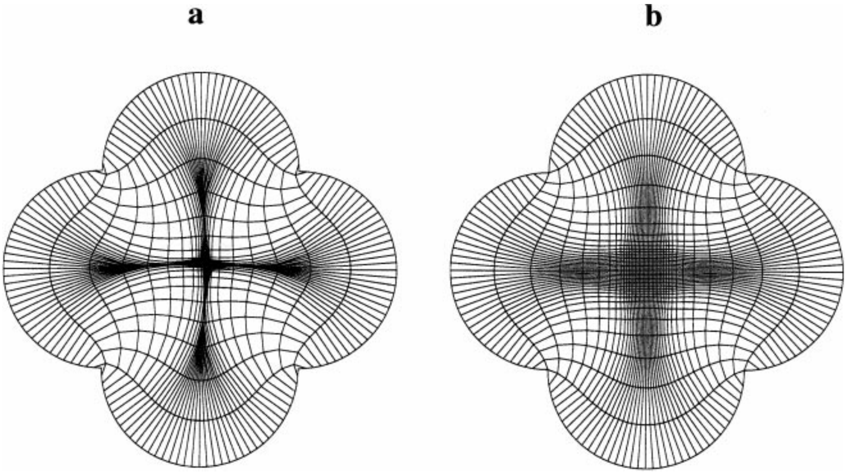

FIG. 10. Domain F, limited by four half-circles on the edges of a unit square. Specified boundaries on all sides. Force constants are 0 and 0.01 in Grids a and b, respectively. Distances between grid points along the boundaries are equal.

6. Domain F (Fig. 10) is constructed by four half-circles around a unit square. Grids are generated with and without force constants and results are compared.

The effects of different force constants and grid point density are investigated using domain A. Results are shown in Figs 11, 12 and 13. Changes of ADO and MDO are shown as functions of grid density (grid points per direction) in Fig. 11. Both MDO and ADO decrease with increasing grid density. In Fig. 12, ADO of grids with different node numbers and force constants are plotted. As expected, with increasing force constant ADO increases. However, the best grid in terms of orthogonality is obtained with a force constant of 0.01, instead of zero. Another characteristic of the method is that, as the force constant increases, the mean aspect ratio AAR decreases, as shown in Fig. 13. The method starts to resemble conformal mapping. This can be seen in terms of mean deviation from orthogonality ADO

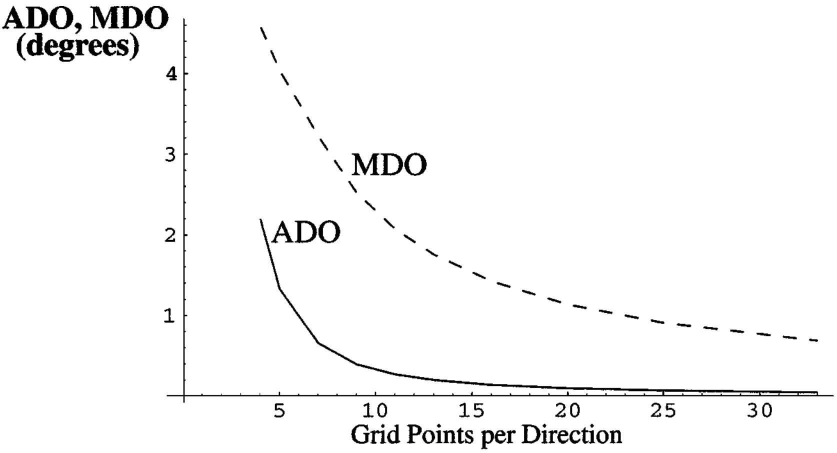

FIG. 11. Maximum and mean deviations from orthogonality (MDO and ADO) with force constant $c = 0.01$ for Domain A.

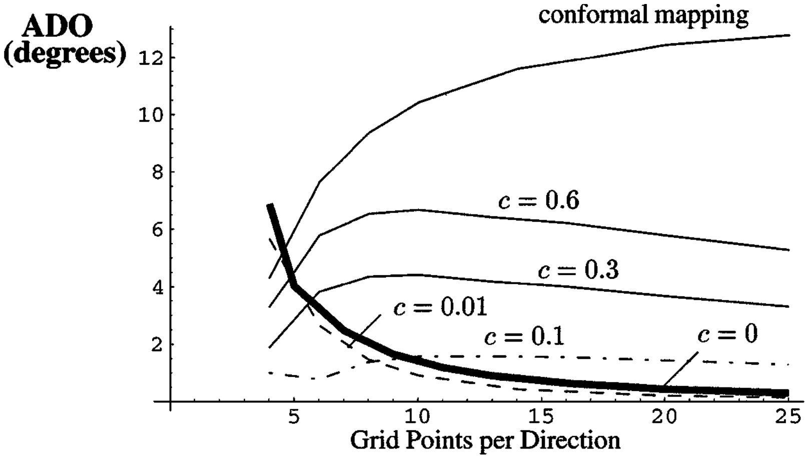

FIG. 12. Mean deviation from orthogonality (ADO) with different force constants for Domain A.

and mean aspect ratio AAR in Figs. 12 and 13. Overall, the number of convergence iterations decreases as the force constant increases.

Domain B is tested for four different situations (Figs. 4a, 4b, 4c, and 4d). In Grid 4a, specified (Dirichlet) boundary conditions are used with zero force constant. Grids 4b and 4d are similar to Grid 4a, but with force constants of 0.01 and 0.11, respectively. In Grid 4c, the force constant is 0.11 and the distance of the first grid point from the boundary is specified on all the boundaries. Grids are 41 × 41 for these four test cases. Results are given in Table I. Similar observations as for domain A can be deduced.

Domain C is used by Oh and Kang [15]. Grids obtained with both specified boundaries and with sliding (Neumann–Dirichlet) boundaries on the bottom are shown in Figs. 5a and 5b. The force constants are 0.01 for both cases. The effects of using sliding boundaries only on the bottom (Grid 6a) are compared with sliding boundaries both on the bottom and top (Grid 6b). Force constants are taken as 0.11 for both these cases. Results can be seen in Table I. For this domain, grid properties (MDO, ADO, MAR, AAR) as functions

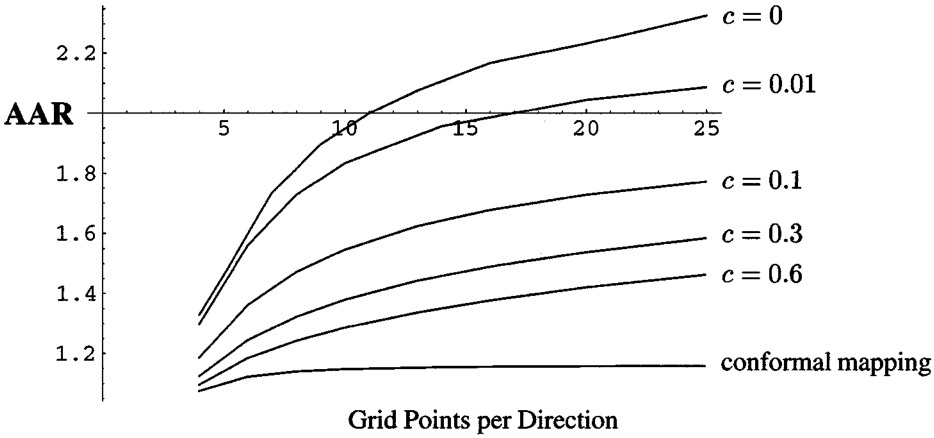

FIG. 13. Mean aspect ratio (AAR) with different force constants for Domain A.

TABLE I
Results for Selected Domain

Domain	Figure	Grid	Force constant c	MDO	ADO	MAR	AAR
A	3a	17 × 17	0.00	6.94	0.57	3.34	2.18
	3b	17 × 17	0.10	6.94	1.5	2.92	1.69
B	4a	41 × 41	0.00	2.26	0.07	N.A.	N.A.
	4b	41 × 41	0.01	0.34	0.08	7.3	1.88
	4c	41 × 41	0.11	19.28	0.74	5.48	1.78
	4d	41 × 41	0.11	1.1	0.49	4.84	1.75
C	5a	21 × 21	0.01	1.68	0.42	12.43	2.73
	5b	21 × 21	0.01	1.81	0.31	10.29	2.64
	6a	21 × 21	0.11	3.32	1.21	6.71	2.25
	6b	21 × 21	0.11	11.65	1.09	9.12	1.72
D	7a	11 × 11	0.001	30.18	6.96	127.64	21.74
	7b	11 × 11	0.1	39.09	12.88	9.01	3.6
	8a	41 × 41	0.1	33.00	2.64	34.52	5.05
	8b	41 × 41	0.1	1.02	0.35	7.66	3.44
E	9b	41 × 41	0.01	3.62	0.92	22.3	3.4
	9c	41 × 41	0.01	0.5	0.04	6.13	2.26
F	10b	41 × 41	0.01	23.96	0.62	14.82	4.2

of force constant are also plotted. Figures 14a and 15a display the case with specified boundaries on all sides. Figures 14b and 15b show grids with specified boundaries at three sides and a sliding boundary on the bottom. For this domain, introducing a small force constant decreases maximum and mean deviation from orthogonality. Another observation from Figs. 14 and 15 is that, although the type of boundary condition does not affect MDO

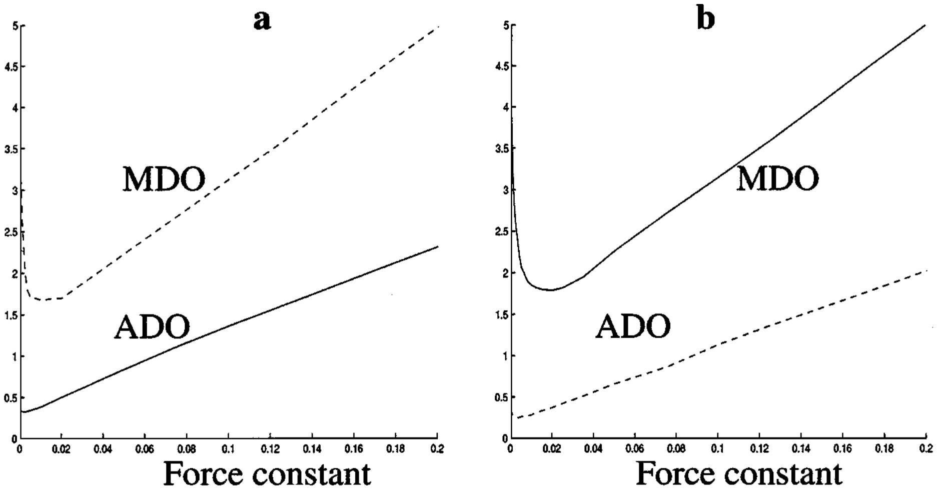

FIG. 14. Maximum deviation from orthogonality (MDO) and mean deviation from orthogonality (ADO) as functions of the force constant for domain C with 21 × 21 grid. In Fig. 14a, specified boundary grid point distribution on all boundaries; in Fig. 14b, sliding boundaries used along $y = 0$.

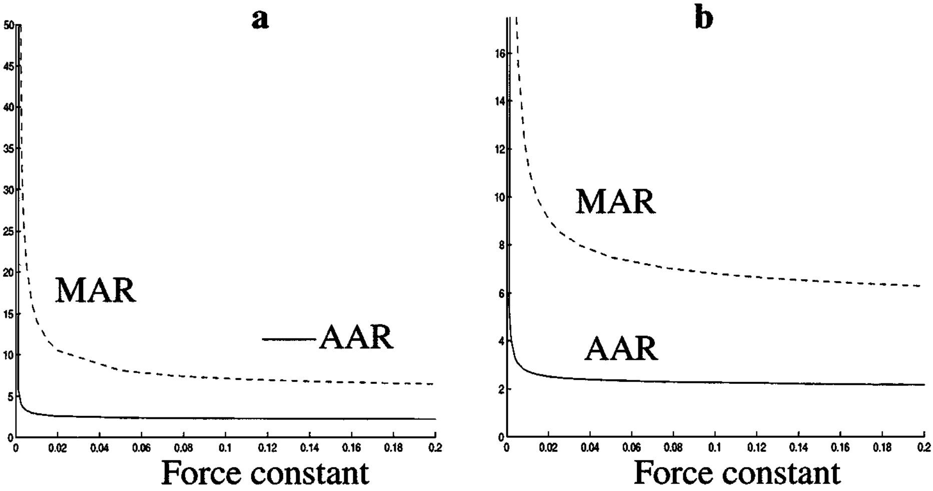

FIG. 15. Maximum aspect ratio (MAR) and mean aspect ratio (AAR) as functions of the force constant for domain C with 21 × 21 grid. Same boundary conditions as in Fig. 14.

and ADO, maximum and mean aspect ratios decrease when a moving boundary is used. Therefore, for this domain, using moving boundaries improves grid quality in terms of aspect ratios.

Test cases using Domain D are illustrated in Figs. 7 and 8. In 7a and 7b, 11 × 11 grids are used. In 8a and 8b, 41 × 41 grids are used. Force constants are 0.001 for Grid 7a, and 0.1 for Grids 7b, 8a and 8b. In Grids 7a, 7b, and 8a, specified boundaries are used on all sides. In Grid 8b, sliding boundaries are used for circular sides and specified at all other sides. Results are shown in Table I. Although mean deviation from orthogonality increases with the force constant, maximum and mean aspect ratios decrease. This is shown in Grids 7a and 7b. MAR and AAR tend to infinity as the force constant approaches zero. Figure 8 illustrates that introduction of Neumann–Dirichlet boundary conditions improves grid quality for this domain.

For Domain E, three different test cases are considered. In Grid 9b specified boundaries are imposed while for Grids 9a and 9c sliding boundaries are used. Force constants are 0 for Grid 9a and 0.01 for Grids 9b and 9c. Grid 9a does not satisfy the convergence criteria. Results are shown in Table I. Figure 9a shows that although the use of sliding boundaries improves grid quality without the use of a force constant, grid lines still collapse onto each other.

Domain F involves two cases. In Grid 10a, the force constant is taken as 0. It is observed that grid lines collapse onto each other. This causes convergence difficulties as well as an unacceptable grid, MAR and AAR approach infinity. In Fig. 10b, a grid is generated for the same domain but with a force constant of 0.01. This eliminates problems with aspect ratios and convergence. Results can be seen in Table I.

Finally, the total number of iterations for convergence is examined as a function of force constant. The number of iterations for convergence decreases with increasing force factor in all grid generation examples. By increasing force constant, the generating system becomes better posed. Numerical evidence is shown in Fig. 16 for domain A. Here, a 41 × 41 grid is used with specified boundary conditions.

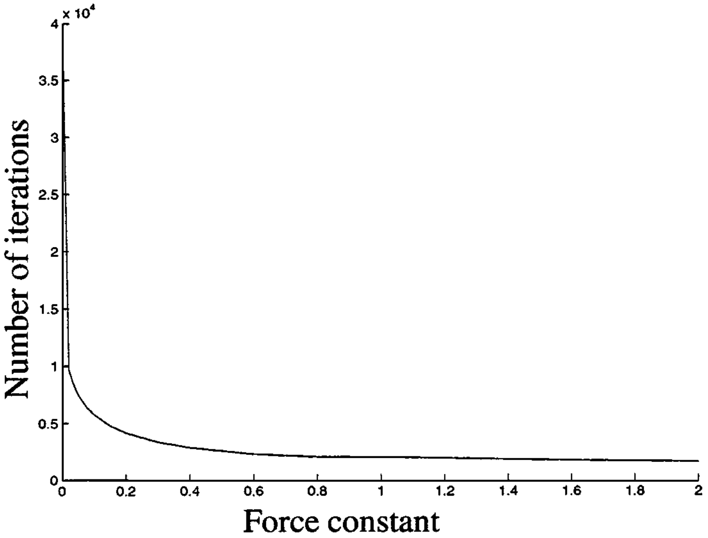

FIG. 16. Number of iterations as a function of force constant for domain A with 41 × 41 grid. Specified boundary grid point distribution on all the boundaries.

6. DISCUSSION AND CONCLUSION

The covariant Laplace operator is frequently used in orthogonal grid generation, with the aim of producing a robust and efficient grid generation algorithm. The method proposed and studied here also uses the covariant Laplace operator. It has several advantages compared to previous methods. Since specification of the distortion function is not required, and the distortion function is instead determined by the domain, the method is more effective and robust. Moreover, it does not suffer the problem of grid lines collapsing onto each other.

The proposed method generates grids having properties between conformal mapping and Eça's method. As the force constant increases, the method resembles conformal mapping. For zero force constant, the method is identical to Eça's method. The latter and similar methods can, for certain domains, produce grids in which the grid lines collapse onto each other, especially when sliding (Neumann–Dirichlet) boundary conditions are used. This was seen in several of the examples. In the proposed method, this problem is eliminated by the introduction of repulsive/attractive force between grid points. Conformal mapping suffers from insufficient robustness and adaptability to boundary conditions. The proposed method generates a grid that incorporates advantages of both approaches. Selection of the force constant allows tuning the grid properties and choosing the relative importance of grid orthogonality and aspect ratio. By preventing the collapse of grid lines onto one another, convergence problems exhibited by previous methods are avoided. It is found that the number of iterations for convergence decreases as the force factor increases. Moreover, as the force constant increases, the mean deviation from orthogonality increases also. In some cases, introducing a small amount of force decreases mean deviation from orthogonality. On the other hand, the mean aspect ratio decreases with increasing force. In the implementation of the method, the force constant can be interactively changed, so that grid properties can be adjusted according to the domain. In this study, the force constant is fixed to the same value throughout the domain. There are no restrictions on its distribution, however, and the force constant can be tailored to specific domains by varying it over the grid.

The application of the method to certain difficult domains showed that using sliding boundaries increases the method's adaptability to boundary conditions considerably. However, using such boundaries in some domains may cause accumulation of grid points on portions of the boundary. For this reason it may not be possible to use sliding boundaries for some domains. To prevent accumulation on the boundaries, controlling forces similar to those applied to the interior grid points can be applied to the boundary nodes. Using sliding boundaries enables boundary grid points to arrange themselves according to orthogonality characteristics of the domain.

REFERENCES

1. J. E. Castillo, *Mathematical Aspects of Numerical Grid Generation* (SIAM, Philadelphia, 1991).
2. J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin, *Numerical Grid Generation* (North-Holland, New York, 1985).
3. D. C. Ives and R. M. Zacharias, Conformal mapping and orthogonal grid generation, *J. Propulsion* **5**, 327 (1989).
4. T. K. Hung and T. D. Brown, An implicit finite-difference method for solving the Navier-Stokes equation using orthogonal curvilinear coordinates, *J. Comput. Phys*. **23**, 343 (1977).
5. S. B. Pope, The calculation of turbulent recirculating flows in general orthogonal coordinates, *J. Comput. Phys*. **26**, 197 (1978).
6. H. J. Hansling and R. M. Coleman, A method for generation of orthogonal and nearly orthogonal boundary-fitted coordinate systems, *J. Comput. Phys*. **43**, 373 (1980).
7. C. D. Mobley and R. J. Stewart, On the numerical generation of boundary-fitted orthogonal curvilinear coordinate systems, *J. Comput. Phys*. **34**, 124 (1980).
8. G. Ryskin and L. G. Leal, Orthogonal mapping, *J. Comput. Phys*. **50**, 71 (1983).
9. M. R. Albert, Orthogonal curvilinear coordinate generation for internal flows, in *Grid Generation Techniques in Computational Fluid Mechanics* (Pineridge Press, Swansea, 1988).
10. A. Allievi and S. M. Calisal, Application of Bubnov-Galerkin formulation to orthogonal grid generation, *J. Comput. Phys*. **98**, 163 (1992).
11. E. P. Ascoli, D. S. Dandy, and L. G. Leal, On distortion functions for the strong constraint method of numerically generating orthogonal coordinate grids, *J. Comput. Phys*. **72**, 513 (1987).
12. L. Eça, 2D orthogonal grid generation with boundary point distribution control, *J. Comput. Phys*. **125**, 440 (1996).
13. E. D. Chikhliwala and Y. C. Yortsos, Application of orthogonal mapping to some two-dimensional domains, *J. Comput. Phys*. **57**, 391 (1985).
14. I. S. Kang and L. G. Leal, Orthogonal grid generation in a 2D domain via the boundary integral technique, *J. Comput. Phys*. **102**, 78 (1992).
15. H. J. Oh and I. S. Kang, A non-iterative scheme for orthogonal grid generation with control function and specified boundary correspondence on three sides, *J. Comput. Phys.* **112**, 138 (1994).
16. R. Duraiswami and A. Prosperetti, Orthogonal mapping in two dimensions, *J. Comput. Phys*. **98**, 254 (1992).
17. T. G. Theodoropoulos and G. C. Bergeles, A Laplacian equation method for numerical generation of boundary-fitted 3D orthogonal grids, *J. Comput. Phys*. **82**, 269 (1989).
18. P. Tamamidis and D. N. Assanis, Generation of orthogonal grids with control of spacing, *J. Comput. Phys.* **94**, 437 (1991).
19. S. Kawata, N. Takura, and Y. Manabe, Grid generation with orthogonality and uniformity of line-spacing changing ratio, *Comput. Phys*. *Commun*. **94**, 19 (1996).
20. K. L. Lin and H. J. Shaw, Two-dimensional orthogonal grid generation techniques, *Comput. Struct*. **41**, 569 (1991).
21. I. S. Sokolnikoff, *Tensor Analysis* (Wiley, New York, 1951).