JOURNAL OF COMPUTATIONAL PHYSICS 94, 437–453 (1991)

Generation of Orthogonal Grids with Control of Spacing

P. TAMAMIDIS AND D. N. ASSANIS

Department of Mechanical and Industrial Engineering,
University of Illinois at Urbana-Champaign, Urbana, Illinois 61801

Received June 7, 1989; revised March 9, 1990

A methodology for generating orthogonal curvilinear grids is applied to two-dimensional domains. An important feature of the methodology is its ability to control effectively the grid spacing, especially near the boundaries. This paper summarizes the governing equations used for grid generation. Then the numerical procedure is described, with special emphasis on the scheme used to enhance stability and accuracy of the solution. The significance of the distortion function and the way it is used to control grid spacing are illustrated in geometries commonly found in engine combustion chambers. The influence of various parameters, including number of grid points, relaxation factor, and range of values of the distorion function, on the performance of the method are also investigated. It is concluded that the methodology can successfully produce smooth orthogonal grids with control of spacing in symmetric and nonsymmetric domains. © 1991 Academic Press, Inc.

1. INTRODUCTION

Numerical generation of grids has recently found extensive application to problems in physics and engineering. This is because many problems of practical interest, such as flows bounded by curved surfaces, can be calculated in curvilinear coordinates more accurately and conveniently than in Cartesian coordinates. In flow problems, the boundary conditions have to represented accurately in the numerical formulation since the region in the immediate vicinity of solid surfaces is generally dominant in determining the character of the flow, especially at high Reynolds numbers.

Thompson *et al.* [1–3] have introduced a powerful approach for the numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies. However, the Thompson *et al.* method has the drawback of yielding a nonorthogonal coordinate system. When the Navier–Stokes equations are solved in nonorthogonal grids, many additional terms arise in the computer code. Thus, programming becomes more difficult, while storage and time requirements increase considerably; also, instability and accuracy problems associated with the existence of metrics in the equations could arise during the solution.

0021-9991/91 $3.00

Some of these problems could be alleviated when the equations are solved in orthogonal curvilinear grids. Barfield [4] proposed a method based on conformal mapping of the boundary of a closed two-dimensional region on the perimeter of a rectangular polygon in which an orthogonal rectilinear grid is inscribed. Meyder [5] presented a method to generate orthogonal curvilinear coordinates by solving for the potential and force lines in a simply-connected region and taking these as the coordinate lines. However, the power of such conformal mappings is limited due to the simultaneous requirements of orthogonality and equality of the scale factors.

Hung and Brown [6], and Pope [7] have used methods less restrictive than the conformal mappings presented in [4, 5]. Specifically, they have considered the ration of scale factors constant throughout the domain, but not necessarily equal to unity. Habib and Whitelaw [8] also used the method of Pope [7] to construct orthogonal grids. Mobley and Stewart [9] have suggested construction of an orthogonal mapping by nonuniform stretching of the conformal coordinates. Another attempt to produce orthogonal curvilinear grids was that of Haussling and Coleman [10]. However, although their method was applied to smooth geometries, in most cases the constructed grids were not orthogonal.

Ryskin and Leal [11] have proposed a flexible methodology for the numerical generation of orthogonal mappings, with the ratio of scale factors variable throughout the domain. They identified three types of problems:

(1) The shape of the domain is not known in advance, but is to be determined as a part of the solution of a physical problem (e.g., free boundary problems in fluid mechanics).

(2) The shape of the domain is known, but the distribution of the coordinate nodes along the boundary is not specified and may be determined by the mapping.

(3) The shape of the domain is known and the distribution of the coordinate nodes is specified along all boundaries; i.e., the complete boundary correspondence is prescribed.

Two different methods of implementing their mapping, i.e. the strong constraint method and the weak constraint method, were presented in relatively simple geometries. The strong constraint method was applied to problems of type (1), while the weak constraint method was applied to problems of type (3). However, neither method was aplied to the study of problems of type (2). Instead, it was simply stated, without giving any details, that the strong constraint method might, in principle, be applied to problems of type (2). It was also suggested, that the weak constraint method could be used conveniently for problems of type (2) by prescribing some reasonable boundary correspondence. Since a reasonable boundary correspondence may not be obvious in most cases, several iterative guesses would be required, thus making the weak constraint method rather impractical for complex problems of type (2).

Chikhliwala and Yortsos [12] investigated further the applicability of the weak constraint method proposed by Ryskin and Leal [11]. They reported that a linear distribution of the shape factor and symmetric geometries resulted in grids of higher accuracy. In contrast, the method did not produce grids of satisfactory accuracy in nonsymmetric regions. No control of grid spacing was reported in their examples. However, the shape factor was prescribed at the boundaries, and according to Ryskin and Leal [11] this should have affected the grid spacing.

The work presented here aims to extend the methodology of Ryskin and Leal [11] in order to solve problems where the shape of the domain is known and the distribution of the coordinate nodes is prescribed to a portion of the boundary where the distortion function is not prescribed. Our methodology can also address problems where the distribution of the coordinate nodes is not prescribed at all on the boundaries of the domain. Furthermore, our ability to prescribe the distortion function in those boundaries where the distribution of the nodes is not prescribed allows for a higher degree of control of grid spacing, especially near the boundaries.

Control of grid spacing is an important requirement since we intend to use the grids for problems arising in the study of reciprocating internal combustion engines. Such typical problems are the calculation of fluid flow through the inlet ports and within the cylinder of internal combustion engines, and the study of transient heat transfer from the combustion gas to the surrounding surfaces [14]. Both problems involve large velocity and temperature gradients near the boundaries. In addition, the geometries of interest are complex, and typically not symmetric.

This paper summarizes the governing equations used for grid generation and the numerical procedure used to solve them. Particular emphasis is placed on adapting a numerical scheme able to achieve stable convergence with minimal numerical error. The ability of the method to control grid spacing will be illustrated in geometries commonly found in engine combustion chambers. Finally, the influence of various parameters, including number of grid points, relaxation factor, and range of values of the distortion function, on the performance of the method will be investigated.

2. Grid Generation Procedure

A. *Mathematial Development*

This section contains the governing equations for generating an orthogonal grid. Since we have adapted the basic system of equations proposed by Ryskin and Leal [11] for our work, only a short description of the formulation will be given here.

The system of equations defining the mapping is produced by observing that x, as a Cartesian coordinate in the physical space, is a linear scalar function of position and the same applies to y. Therefore, grad(x) and grad(y) are constant valued vector fields, and it follows that

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

where ∇^2 is the covariant Laplace operator given by

$$\nabla^2 = \frac{1}{h_\xi h_\eta}\left\{\frac{\partial}{\partial \xi}\left(f\frac{\partial}{\partial \xi}\right)+\frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial}{\partial \eta}\right)\right\}.$$

Equation (1) can be written in the orthogonal ξ, η coordinates as

$$\frac{\partial}{\partial \xi}\left(f\frac{\partial x}{\partial \xi}\right)+\frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial x}{\partial \eta}\right)=0 \tag{2}$$

$$\frac{\partial}{\partial \xi}\left(f\frac{\partial y}{\partial \xi}\right)+\frac{\partial}{\partial \eta}\left(\frac{1}{f}\frac{\partial y}{\partial \eta}\right)=0. \tag{3}$$

In the above equations, $f(\xi, \eta)$ is the distortion function, which is defined as the ratio of the scale factor in η-direction, over the scale factor in ξ-direction; i.e.,

$$f(\xi, \eta) = h_\eta / h_\xi, \tag{4}$$

where h_η and h_ξ are given by

$$\left(\frac{\partial h}{\partial \eta}\right)^2 = \left(\frac{\partial x}{\partial \eta}\right)^2 + \left(\frac{\partial y}{\partial \eta}\right)^2$$

$$\left(\frac{\partial h}{\partial \xi}\right)^2 = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2.$$

Using the fact that in an orthogonal grid the metric g_{12} given by

$$g_{12} = x_\xi x_\eta + y_\xi y_\eta$$

is zero, two additional equations have to be valid everywhere in the domain:

$$f\frac{\partial x}{\partial \xi} = \frac{\partial y}{\partial \eta}, \qquad \frac{\partial x}{\partial \eta} = -f\frac{\partial y}{\partial \xi}. \tag{5}$$

For the special case of $f(\xi, \eta) = 1$, Eqs. (5) become the Cauchy-Riemann equations which are used in conformal mapping. We also abserve that Eqs. (5) satisfy both Eqs. (2) and (3). In fact, by suitably differentiating Eqs. (5), we can produce another system of differential equations which could be used instead of Eqs. (2) and (3) for grid generation.

Since our objective is to generate a grid with control of spacing, an additional aquation for the distortion function is needed to complete the formulation. Our methodology for controlling grid spacing differs considerably from the procedure in [11]. Thus, it will be discussed in detail in the following section of the paper.

B. *The Distortion Function*

First, let us explain the physical meaning of the distortion function defined by Eq. (4). The distortion function specifies the ratio of the sides of a small rectangle in the x, y plane which is mapped onto a rectangle in the ξ, η plane. The values that f can take are always positive with a reasonable upper limit of about 10, as can be concluded from Fig. 1. If the numerical values of f throughout the domain could be controlled, this would enable us to control the spacing of the nodes everywhere in the domain. For example, to attract the grid lines towards a boundary where $\eta = \text{const.}$ then, simply, f values less than unity should be prescribed at the nearby points. On the other hand, attraction towards a boundary where $\xi = \text{const.}$ will be achieved when f greater than unity is prescribed. This shall be illustrated in Section 3, where several numerical examples will be given.

In order to solve the system of Eqs. (2) and (3) for the values of the coordinates x, y at a given grid point, the value of the distortion function f at that point needs to be known. As will be explained in Section C, $f(\xi, \eta)$ cannot be prescribed from the beginning everywhere in the domain, without possibly violating the orthogonality condition close to the boundaries. Instead, the variation of f in the domain should be obtained from the solution of a differential equation. We propose an equation of the form

$$\frac{\partial^2 f}{\partial \xi^2} + \frac{\partial^2 f}{\partial \eta^2} = G(\xi, \eta). \tag{6}$$

The motivation for suggesting Eq. (6) is that it could control grid spacing and provide a smooth variation of f throughout the domain. The latter is a very critical requirement to minimize deviations from orthogonality. Ryskin and Leal [11] had earlier suggested that an elliptic partial differential equation (of the form of Eq. (6) but with the RHS equal to zero) could be used. However, instead of demonstrating the use of an elliptic PDE they preferred to use a simpler algebraic approach for their studies. In our opinion, their approach may have merits for grid generation in smooth peanut-shaped domains; however, in the course of our study, it was found that the algebraic scheme would give unsatisfactory results when applied to nonsmooth geometries. Thus, Poisson's equation (6) was chosen.

The function $G(\xi, \eta)$ in Eq. (6) can be used to control the distribution of the

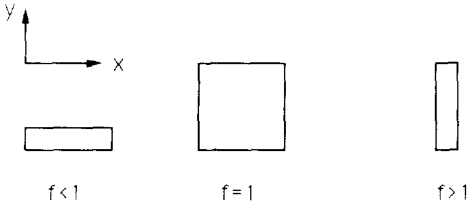

FIG. 1. Graphical representation of three elements with the corresponding value of f for each element.

distortion function throughout the domain; this is equivalent to controlling the grid spacing. Thus, $G(\xi, \eta)$ is given the name "generation of distortion." $G(\xi, \eta)$ should typically involve some combination of sinusoidal, cosinusoidal and exponential functions, i.e.,

$$G(\xi, \eta) = \text{const.} \cdot f_1(\xi) \cdot f_2(\eta), \tag{7}$$

where const. is a suitable real number, $f_1(\xi)$ is a function of ξ, and $f_2(\eta)$ is a function of η; the only requirement is that Eq. (6) give positive values of f everywhere in the domain. Another treatment is to write G in the form

$$G(\xi, \eta) = C(\xi, \eta) + V(\xi, \eta) \cdot f(\xi, \eta). \tag{8}$$

In usual problems, we set $V(\xi, \eta) = 0$ everywhere, so that $C(\xi, \eta) = G(\xi, \eta)$. It turns out that Eq. (8) can be used to give adaptive features to our grid generation. This can be recognized by observing that giving large values to $C(\xi, \eta)$ and $V(\xi, \eta)$, say $\pm 10^{30}$, the solution of Eq. (6) at a point yields the prescribed value of f at that point.

Equations (2) and (3) are linked together via the distortion function Eq. (6) to form a system of three simultaneous partial differential equations. Solution of the system of PDEs with appropriate boundary conditions will yield the coordinates af an orthogonal grid with control of spacing. At each boundary we can apply the following conditions for $x(\xi, \eta)$ or $y(\xi, \eta)$:

(i) Dirichlet condition, i.e., prescribed value of the variable at the boundary.

(ii) Neumann condition, i.e., prescribed normal derivative at the boundary.

(iii) Mixed Dirichlet and Neumann conditions.

The distortion function may or may not be prescribed at the boundary. If f is prescribed, then only the shape of the boundary can be prescribed (but not the distribution of x and y). If f is not prescribed, its value at the boundary can be found using Eq. (5).

C. *Numerical Procedure*

In this section, we will summarize the numerical procedure used to solve the governing equations of the grid generation. Emphasis will be placed on presenting the numerical scheme developed to achieve stable convergence and minimize inaccuracies during the solution process.

The standard approach of discretizing Eqs. (2) and (3) is by using central differences, second order in accuracy. Since the same procedure applies to both equations, we will present the procedure only for Eq. (2). Equation (2) can be rewritten as

$$f^2 f_\xi x_\xi + f^3 x_{\xi\xi} - f_\eta x_\eta + f x_{\eta\eta} = 0. \tag{9}$$

Using the second-order finite difference representations,

$$\begin{aligned}
x_\xi &= (x_\mathrm{E} - x_\mathrm{W})/(2\Delta\xi) \\
x_\eta &= (x_\mathrm{N} - x_\mathrm{S})/(2\Delta\eta) \\
x_{\xi\xi} &= (x_\mathrm{E} - 2x_\mathrm{P} + x_\mathrm{W})/(\Delta\xi)^2 \\
x_{\eta\eta} &= (x_\mathrm{N} - 2x_\mathrm{P} + x_\mathrm{S})/(\Delta\eta)^2
\end{aligned}$$

with $\Delta\xi = \Delta\eta$, Eq. (9) becomes

$$a_\mathrm{P} x_\mathrm{P} = a_\mathrm{E} x_\mathrm{E} + a_\mathrm{W} x_\mathrm{W} + a_\mathrm{N} x_\mathrm{N} + a_\mathrm{S} x_\mathrm{S}. \tag{10}$$

The notation (P, E, W, N, S) stands for (i, j), $(i+1, j)$, $(i-1, j)$, $(i, j+1)$, $(i, j-1)$ points, respectively. The coefficients a_i are

$$\begin{aligned}
a_\mathrm{P} &= a_\mathrm{E} + a_\mathrm{W} + a_\mathrm{N} + a_\mathrm{S} \\
a_\mathrm{E} &= f^3 + 1/2 f^2 f_\xi \\
a_\mathrm{W} &= f^3 - 1/2 f^2 f_\xi \\
a_\mathrm{N} &= f - 1/2 f_\eta \\
a_\mathrm{S} &= f + 1/2 f_\eta.
\end{aligned}$$

Clearly, under certain conditions some a's could be positive and other negative. This situation would result in unstable convergence, or inaccurate results. This is most probable when initial guesses for the x, y coordinates are very different from their final values, or when we try to make the grid finer in some areas. We believe that the poor results reported in [12] for nonsymmetrical domains are in part due to such reasons. The same reasons may account for the poor results of [10], as well. Unfortunately, the existence of first derivatives in Eqs. (2) and (3) causes many problems.

Since equations for grid generation with only second derivatives have yet to be proposed, the existing numerical scheme can be improved by ensuring that all coefficients in Eq. (10) have the same sign. Thus, we use forward finite differences for x_ξ when f_ξ is positive, and backward differences when f_ξ is negative. On the other hand, backward finite differences are used for x_η when f_η is positive, and forward differences when f_η is negative. Hence, the coefficients in Eq. (10) become

$$\begin{aligned}
a_\mathrm{P} &= a_\mathrm{E} + a_\mathrm{W} + a_\mathrm{N} + a_\mathrm{S} \\
a_\mathrm{E} &= f^3 + f^2 [f_\xi, 0] \\
a_\mathrm{W} &= f^3 + f^2 [-f_\xi, 0] \\
a_\mathrm{N} &= f + f^2 [-f_\eta, 0] \\
a_\mathrm{S} &= f + [f, 0].
\end{aligned} \tag{11}$$

The symbol $[\alpha, \beta]$ denotes the greater of α and β. It should be mentioned that fourth-order finite differences have been used for f_ξ and f_η to improve the performance of the numerical scheme.

The same procedure is followed for Eq. (3) yielding a discretization equation for y of the form

$$a_P y_P = a_E y_E + a_W y_W + a_N y_N + a_S y_S, \tag{12}$$

where the coefficients are defined as those of Eq. (11). The discretization of Eq. (6) is straightforward, and shall not be discussed here.

The choice of $\Delta\xi = \Delta\eta$ in our finite difference scheme has important implications for the distortion function. If M points in the x direction and N points in the y direction were used to map the domain in the x, y plane onto a rectangular region in the ξ, η plane, the ratio of the sides of the rectangle would be M/N. However, the latter ratio cannot be arbitrarily chosen, because it depends upon the domain and the distortion function [13]. If f were chosen a priori, then we would either have to drop the orthogonality condition, or to update the values of $\Delta\xi$ and $\Delta\eta$ (which would no longer be equal to each other) at every iteration. The latter strategy would increase substantially the programming effort as well as the computational time. Thus, f should only be prescribed at the boundaries of interest, and not everywhere in the domain.

Since the distortion function cannot be prescribed a priori everywhere in the domain, an iterative procedure must be used to solve the system of PDEs that govern grid generation. A suitable scheme is presented below:

(1) Initialize $x(\xi, \eta)$ and $y(\xi, \eta)$ at the portion of the boundary where f is prescribed. This has been implemented by first guessing $x(\xi, \eta)$ and then using cubic-spline polynomials to find $y(\xi, \eta)$.

(2) Solving the PDEs

$$\frac{\partial^2 x}{\partial\xi^2} + \frac{\partial^2 x}{\partial\eta^2} = 0; \qquad \frac{\partial^2 y}{\partial\xi^2} + \frac{\partial^2 y_2}{\partial\eta}^2 \approx 0, \tag{13}$$

initialize the distributions of $x(\xi, \eta)$ and $y(\xi, \eta)$. The resulting grid is curvilinear.

(3) Using one of Eqs. (5), or Eq. (4), find the distortion function at the portion of the boundary where it is not prescribed. In the remaining of the boundary, use the prescribed values of f.

(4) Once f is known at the entire boundary, use Eq. (6) to find f in the interior of the domain.

(5) Solve Eqs. (2) and (3) using the ADI method (alternating direction iteration) with the TDMA algorithm.

(6) Apply the boundary conditions for x and y in order to find their distribution at the portion of the boundary where they are not prescribed.

(7) Check if convergence has been attained. If not, go to step 2 and repeat.

3. Illustration of the Method

A. *Examples of Grid Generation*

In this section we shall present the results obtained using the proposed method for two geometries of interest. First, an example of the fluid domain in an engine combustion chamber with a bowl-in piston is given in Fig. 2. In this case, 31 × 21 grid points were used, and the maximum deviation angle from orthogonality (MDO) was found to be 1.453°. The average deviation angle from orthogonality (ADO) calculated using deviation errors at every point of the domain is 0.35°. Both MDO and ADO values are very small, considering the difficult geometry of the domain. Overall, the generated grid is very smooth. This is a very critical requirement for any type of grid, orthogonal or nonorthogonal, especially at regions where large gradients prevail. Even when robust numerical techniques such as the finite-volume method are used, a smooth grid would give more accurate results as discussed in [15].

The boundary conditions for x and y were

$$\begin{array}{lll}
\text{side } AB & y = y(x), & \dfrac{\partial x}{\partial \eta} = -f(\xi, \eta)\dfrac{\partial y}{\partial \xi} \\[2ex]
\text{side } BC & x = L, & \dfrac{\partial y}{\partial \xi} = 0 \\[2ex]
\text{side } CD & y = 0 & \dfrac{\partial x}{\partial \eta} = 0 \\[2ex]
\text{side } DA & x = 0 & \dfrac{\partial y}{\partial \xi} = 0.
\end{array}$$

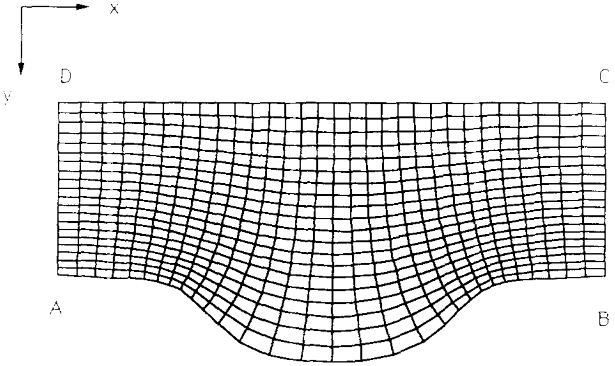

FIG. 2. Orthogonal grid. $f = 0.5$ on side AB, $G = 0$, everywhere. 31 × 25 points.

For the distortion function the following boundary conditions were used:

side AB. f was prescribed equal to 0.5.

sides BC, CD, and DA. f was not prescribed, but was calculated at each iteration using Eq. (4).

As shown in Fig. 2, by specifying $f = 0.5$ on side AB, we have constructed a well-proportioned orthogonal grid. However, we may want to attract the grid lines towards side AB in order to calculate more accurately the fluid motion directly above the piston (where large velocity gradients are likely to exist). One way of achieving this objective is by prescribing a smaller value of f on side AB. According to Section 2B, this will result in a finer grid near that side. To illustrate this, the same geometry and boundary conditions discussed previously have been used, but $f = 0.1$ was prescribed on side AB. The grid constructed is shown in Fig. 3. Indeed, a smaller value of f at side AB results in attracting the grid towards that side. For that grid, MDO and ADO are 5.24 and 1.45°, respectively. Although MDO and ADO have been increased by decreasing f, the resulting grid is still satisfactory. As discussed in [15], a value of ADO equal to 2° is the upper limit for the grid to be considered orthogonal. For ADOs smaller than 2°, the magnitudes of the additional terms arising due to the non-orthogonality of the grid are small. However, for ADOs larger than 2°, the grids should be treated as nonorthogonal and non-orthogonal flow codes would have to be used.

Now the question is how the grid can be attracted even more towards side AB. For reasons that will be explained in Section 3B, further decreasing (below 0.1) the value of f on side AB, will not improve the grid attraction towards that side. Instead, an effective way of increasing grid density is by prescribing a nonzero $G(\xi, \eta)$ in Eq. (6). Note that in the two previous examples, G was set to zero everywhere. Keeping every other parameter the same as before, we additionally prescribe

$$G(\xi, \eta) = 0.01 \cdot \sin(\eta\pi/2) \cdot |\cos(\pi\xi)|,$$

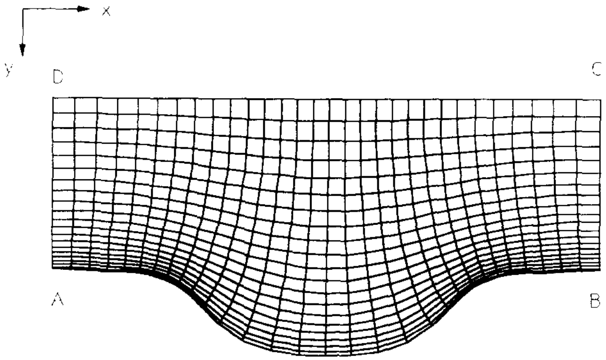

FIG. 3. Orthogonal grid 31 × 25 nodes. $f = 0.1$ on side AB, $G = 0$ everywhere.

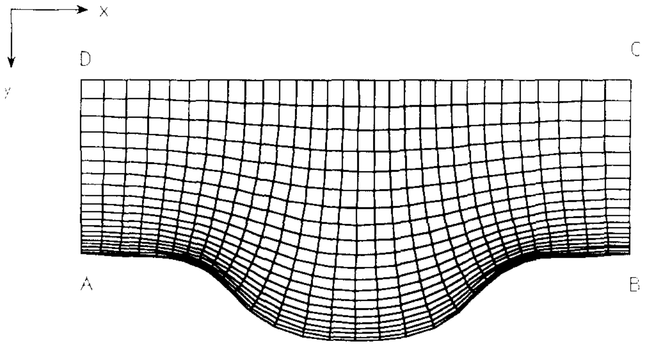

FIG. 4. Orthogonal grid, 31 × 25 nodes. $f = 0.1$ on side AB, $G(\xi, \eta) = 0.01 \cdot \sin(\pi\eta/2) \cdot |\cos(\pi\xi)|$.

where ξ varies from zero on side AD to one on side BC, and η varies from zero on side DC to one on side AB. The resulting grid, shown in Fig. 4, has an MDO of 4.91° and an ADO of 1.43°. The smaller values of MDO and ADO (than in the previous example) can be attributed to the smoother variation of f achieved in this case. As desired, the attraction of the grid towards side AB has increased.

Now some insight will be given into how the function G, used in the example of Fig. 4, was chosen. In the particular example, we are interested in attracting the grid nodes towards side AB more than in the example of Fig. 3. Using a greater than zero value of G results in smaller values of f in the domain, which achieves grid attraction. Since a large positive value of G could even result in negative values of f, a scaling constant equal to 0.01 is used for control purposes. The sinusoidal term is used to control the grid spacing in the vertical direction, so as to achieve gradual attraction towards side AB. Note that this term varies gradually from zero on side DC, thus giving zero G on that side, to one on side AB. The cosinusoidal term is intended to vary the vertical grid attraction with horizontal position in the domain, so as to increase the vertical grid attraction close to corners A and B. For G to be positive for values of ξ greater than $\frac{1}{2}$, we have considered only the absolute values of the cosinusoidal term. We should note here, that our choise of G in this example is not the only possible. In principle, any function proposed in [1–3] for

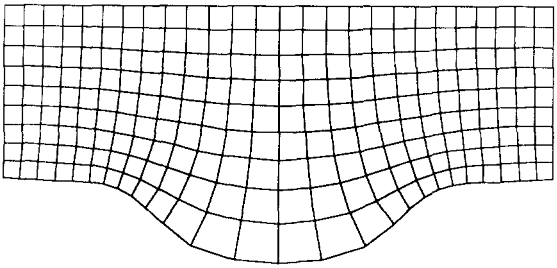

FIG. 5. Conformal mapping.

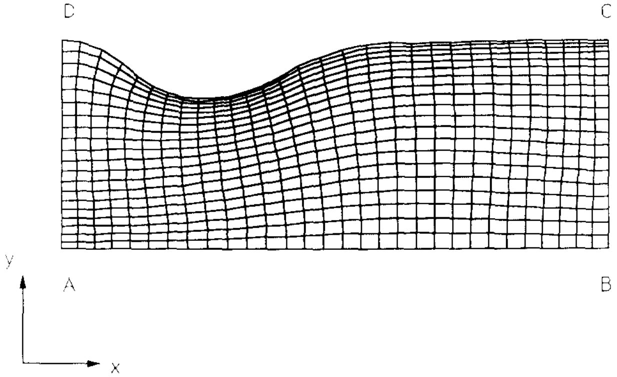

FIG. 6. Orthogonal grid, 31 × 25 nodes.

control of the grid spacing in nonorthogonal grids may be used. However, we must be aware that there is the possibility of obtaining a negative value of f in some points. If this happens, convergence is destroyed. Thus, the G function should be chosen with care.

To demonstrate the advantages of this method over others in controlling the grid spacing, conformal mapping has been used to construct a grid for the same geometry. The resulting grid is presented in Fig. 5. Obviously, a serious drawback of the conformal mapping is that we have no control over the grid spacing.

We shall end this section by presenting a grid generation in a nonsymmetric geometry. An example of the heat conduction domain in an offset bowl-in piston is presented in Fig. 6. To illustrate the ability of our method in attracting the grid towards a desired region of the domain, we have prescribed a nonconstant value for f on boundary DC. In particular, f was varied from 0.5 at point D to 0.1 at point C according to the distribution $f = 0.5 - 0.4\,\xi^{1.2}$. This distribution resulted in attracting the grid towards the right side of the bowl. For this case, MDO and ADO are 4.28° and 0.89°, respectively. The relatively small deviations from orthogonality suggest that the method can be applied with equal success to generate grids in nonsymmetric as in symmetric domains.

B. *Sensitivity Studies*

The performance of any method of constructing orthogonal grids can be primarily evaluated based on three criteria, namely the number of iterations, the maximum and average deviation angles from orthogonality, and the ability of the method to control the spacing over the domain. Additional considerations include whether performance depends on the initial guesses, whether the method can handle geometrical complexities, etc. In this section, the influence of various critical parameters on the performance of our grid generation method will be discussed. As an illustrative example, the geometry presented in Fig. 2 has been chosen.

a. *Control of Spacing*

The ability of the method to control spacing over the domain depends on the geometry of the domain, the initial guesses for the x, y coordinates, the aspect ratio of the domain, the boundary conditions for x, y, the boundary conditions for the distortion function f, and the type of generation of distortion function $G(\xi, \eta)$. The geometry of the domain has been found to affect mostly the orthogonality of the grid, and not the ability to control spacing; the latter is almost independent of geometry.

The initial guesses for x, y has some effect on the ability to control spacing, as well. We have found that with educated initial guesses, construction of the grid is achieved more readily. This is mostly important when the aspect ratio of the domain is substantially different than unity. Also, when we prescribe a Neumann boundary condition for one coordinate and a Dirichlet condition for the other, the control of the spacing can be performed more effectively. The importance of the distortion function $f(\xi, \eta)$ and the generation of distortion function $G(\xi, \eta)$ with regard to grid spacing has been illustrated in Section 3A. The critical requirement for smooth variation of f and G over the domain should be emphasized here. In most cases, the coefficients of the algebraic system that is solved, namely the a's given by Eq. (11), are small numbers, less than unity. Therefore the solution is very sensitive to round-off errors.

b. *Number of Iterations*

The number of iterations depends on the number of nodes, the relaxation factor, and the distribution of the distortion function f. This relation is illustrated in Fig. 7 for different values of f prescribed on side AB. For a given f, the number of itera-

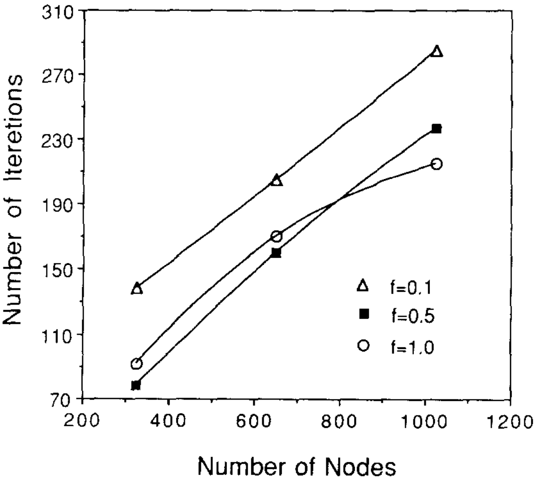

FIG. 7. Effect of the number of nodes and specified f on side AB on the number of iterations.

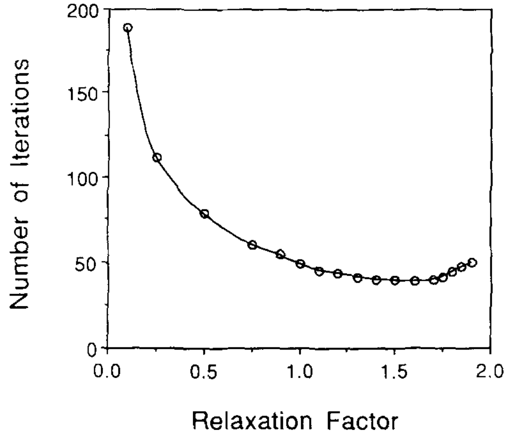

FIG. 8. Number of iterations versus relaxation factor.

tions increases with the number of nodes. Also very small values of f (for instance, $f=0.1$) result on larger number of iterations. This can be attributed to the smaller values of the a_i coefficients of the linear algebraic system. By increasing f, we increase the value of the diagonal coefficients of the tridiagonal matrix, and thus the Gauss-Seidel scheme converges in fewer iterations. However, if we increase f over a critical value (for this case $f=1.0$) which depends upon the geometry and the number of nodes, some nodes (near side AB) effectively get smaller values of f, and thus the number of iterations begins to increase again.

The effect of the relaxation factors on the number of iterations is illustrated in Fig. 8. The relaxation factors for x and y were set equal to each other. Initially, as the relaxation factor increases towards 1.0, the number of iterations decreases significantly. Although a minimum is reached for a relaxation factor equal to 1.6, the improvement is not significant beyond a value of 1.0. For larger values of the relaxation factor (above 1.6), convergence becomes slower again, and ultimately (beyond 2.0 for this geometry) the solution may diverge. Since smaller values of the relaxation factor generally result in more stable convergence, a value of 1.0 (instead of 1.6) was used in our work.

c. *Deviation from Orthogonality*

This section examines the dependence of MDO and ADO on various parameters. Undoubtedly, the most important parameter is the geometry of the domain. We have found that there may be corners and angles on the boundaries in critical areas which increase local deviations from orthogonality, resulting in some high MDOs; however, even for these grids, ADO would generally be satisfactory. Of course the question of whether we can construct an orthogonal grid in any two-dimensional domain still remains. However, we believe that, for most two-dimensional domains, the increasing deviation from grid orthogonality that results with increasing

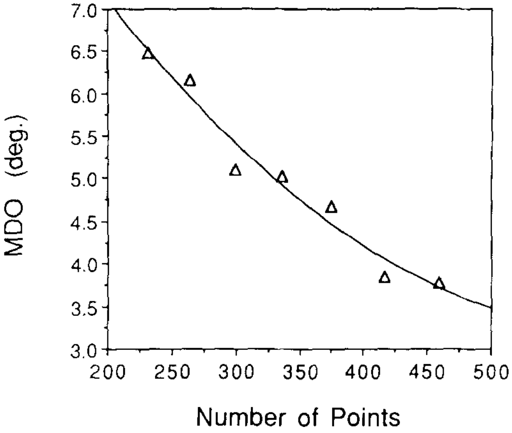

FIG. 9. Maximum deviation from orthogonality versus number of nodes. $f = 0.2$ on side AB.

geometrical complexity is mostly due to numerical errors associated with larger x, y gradients.

Although there is no obvious way of presenting in a graph the dependence between MDO or ADO and geometry, more complex geometries are likely to require a larger number of nodes. The relation between MDO and number of nodes is shown in Fig. 9. The figure suggests that the greater the number of nodes, the less the MDO. This is expected because the transition from continuous to discrete space is done more accurately when a greater number of nodes is used. Figure 9 has been obtained using $f = 0.2$ on side AB.

However, the deviation from orthogonality also depends on the prescribed value of f (and not just on the number of nodes). The effect of the value of the distortion function f on ADO (instead of MDO) is shown in Fig. 10. Initially, as f on side AB

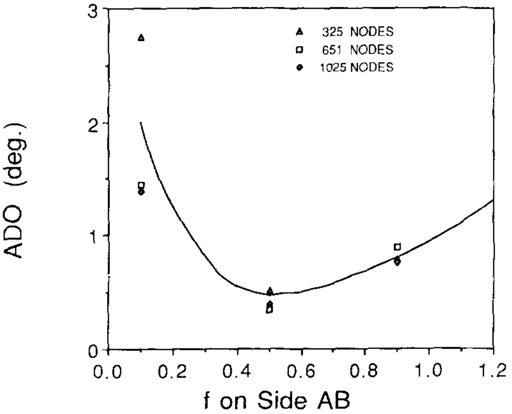

FIG. 10. Average deviation from orthogonality as a function of f on side AB.

increases up to a critical value, the diagonal coefficients of the linear system increase, and the gradients of x and y near the boundary can be calculated more accurately in the ξ, η plane. These two effects result in smaller ADO. However, increasing f on side AB beyond a critical value will result in less accurate calculation of the gradients near side AB, as well as in smaller values of f directly above AB; these effects would increase ADO. The trade-offs involved in the calculation of the gradients result in an optimum value for minimum ADO. In our example, this optimum value of f is approximately equal to 0.5.

4. Conclusions

A method for numerical generation of orthogonal curvilinear grids has been presented. The most attractive feature of the method is its ability to control grid spacing. The numerical scheme used to solve the governing equations has been optimized to ensure stable convergence of the solution with minimal numerical error. Application of the method has been illustrated in geometries commonly found in engine combustion chambers. The influence of various parameters, including number of grid points, relaxation factor, and range of values of the distortion function, on the performance of the method has been investigated. We can conclude the following:

(i) The generated grids are smooth. This is a very critical requirement for any type of grid, orthogonal or nonorthogonal, especially at regions where large gradients prevail.

(ii) The generated grids are orthogonal at the boundaries, as well as in the interior. The orthogonality of the grid at the boundaries is a very attractive feature since normal derivatives at the boundaries can easily be calculated.

(iii) The spacing of the grid lines can be effectively controlled by manipulating the prescription of the distortion function at the boundaries, or the function generating the distortion. Thus, grids can be produced which are finer near the boundaries, or anywhere within the two-dimensional domain where large gradients prevail.

(iv) The methodology can be applied with equal success to generate grids in symmetric and nonsymmetric domains.

Overall, it is anticipated that orthogonal grids with control of spacing generated using the methodology proposed in this paper can be successfully applied to the numerical simulation of fluid flow and heat transfer processes in internal combustion engines.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grants CBT-8858310 and CBT-880002. The National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign provided computer time on the CRAY X-MP/48 system for this study.

REFERENCES

1. F. C. THAMES, J. F. THOMPSON, C. W. MASTIN, AND R. L. WALKER, *J. Comput. Phys.* **24**, 245 (1977).
2. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **24**, 274 (1977).
3. J. F. THOMPSON, Z. V. A. WARSI, AND C. W. MASTIN, *Numerical Grid Generation, Foundations and Applications* (North-Holland, Amsterdam, 1985), p. 188.
4. W. D. BARFIELD, *J. Comput. Phys.* **5**, 23 (1970).
5. R. MEYDER, *J. Comput. Phys.* **17**, 53 (1975).
6. T. K. HUNG AND T. D. BROWN, *J. Comput. Phys.* **23**, 343 (1977).
7. S. B. POPE, *J. Comput. Phys.* **26**, 197 (1978).
8. M. A. HABIB AND J. H. WHITELAW, *Numer. Heat Transfer* **5**, 145 (1982).
9. C. D. MOBLEY AND R. J. STEWART, *J. Comput. Phys.* **34**, 124 (1980).
10. H. J. HAUSSLING AND R. M. COLEMAN, *J. Comput. Phys.* **43**, 373 (1981).
11. G. RYSKIN AND L. G. LEAL, *J. Comput. Phys.* **50**, 71 (1983).
12. E. D. CHIKHLIWALA AND Y. C. YORTSOS, *J. Comput. Phys.* **57**, 391 (1985).
13. L. V. AHLFORS, *Lectures in Quasiconformal Mappings* (Van Nostrand, Princeton, NJ, 1966), p. 8.
14. D. N. ASSANIS AND E. BADILLO, *Trans. Soc. Automat. Eng.* **96**, No. 4, 82 (1987).
15. P. TAMAMIDIS, M. S. thesis, University of Illinois at Urbana-Champaign, 1990 (unpublished).