JOURNAL OF COMPUTATIONAL PHYSICS 39, 164–178 (1981)

An Initial Value Approach to the Production of Discrete Orthogonal Coordinates

CHARLES W. DAVIES

Blackett Laboratory, Imperial College, London SW7, England

Received May 24, 1978; revised August 27, 1979

A numerical procedure is described for the orthogonalization of a discrete non-orthogonal coordinate system in two dimensions. One set of the non-orthogonal coordinate lines is preserved by the orthogonalization procedure; the other set is shifted, by interpolation, to form the new orthogonal system. The procedure is based upon an initial-value formulation with the non-orthogonal coordinates as the independent variables. A first order linear partial differential equation is derived which describes the orthogonal trajectories. A discrete representation of the equation is suggested and an interpolation procedure is described. Examples of non-orthogonal with orthogonalized meshes are given including closed and open coordinate lines. The method may be used to generate boundary fitted coordinates for boundary value problems or Eulerian time dependent codes. It may also be used at each timestep in a waterbag or semi-Lagrangian code.

1. INTRODUCTION

Owing to the high speed of present-day computers, the running of a quite sophisticated two-dimensional finite difference program often uses only a small portion of the computer time available to the user. It is no longer necessary to write a program which will solve the immediate problem in the shortest possible computer time. On the contrary, considering the long times taken to develop and debug such programs, it is well worth ensuring that they have the widest possible applicability, even at the expense of a longer running time. We may practice this principle by writing two-dimensional Eulerian codes which can be used in any boundary-fitted orthogonal coordinate system. This is achieved by expressing the vector differential operators (div, grad, curl, etc.) in their general orthogonal coordinate forms (see, for example, Morse and Feshbach [1]). The particular coordinate system to be used must be supplied in the form of two scale factors for each mesh point. Although some computer time would be wasted, because of the more complex algorithms, when using such a code in cartesian coordinates (both scale factors being constants) the code would simply and accurately deal with complex boundaries. For the new analytic coordinate systems the scale factors are defined by known functions; in the more general case a computational procedure for generating an orthogonal mesh is required.

0021-9991/81/010164–15$02.00/0

Many schemes for producing discrete orthogonal coordinates in two dimensions have already been reported [2–8]. This paper describes a method which, unlike earlier work [2–8], solves a single first order partial differential equation numerically.

The non-orthogonal mesh is described by the cartesian coordinates $x(i, j)$, $y(i, j)$ for each point (i, j). In the continuous limit, corresponding to an infinite number of points, $x(i, j)$ and $y(i, j)$ are parametric forms for the lines $i(x, y) = \text{constant}$ and $j(x, y) = \text{constant}$ (i-lines and j-lines for short). In the continuum, the coordinates are non-orthogonal if

$$\nabla i \cdot \nabla j \neq 0.$$

The discrete mesh is non-orthogonal if the difference form for $\nabla i \cdot \nabla j$, calculated to the chosen accuracy, differs from zero by more than this accuracy.

The orthogonalization is achieved by retaining one set of lines, the j-lines, and moving the i-lines so that $\nabla i \cdot \nabla j$ is zero within the limits of discreteness. The problem is posed as an initial-value problem for the new orthogonal coordinate k with initial conditions equivalent to specifying mesh points on one chosen j-line. A partial differential equation for $k(i, j)$ is derived and written in difference form with the non-orthogonal coordinates (i, j) as the independent variables; the non-orthogonal lines must therefore be smoothly represented by the mesh points.

Because the method preserves the j-lines, and because it is fast, it may be used at every time step in a semi-Lagrangian code in which the j-lines, which would usually correspond to contours of physical interest, move with the fluid whereas the i-lines are defined by the orthogonality condition. This "waterbag" concept is applicable to both hyperbolic and elliptic problems and is reviewed in [11].

Two equivalent equations for the orthogonal trajectories are derived in Section 2. One equation is more suitable than the other and Sections 3 and 4 describe the differencing and interpolation required to produce the orthogonal coordinates. Section 5 summarises the orthogonalization procedure. Examples of open and closed line coordinate systems are given in Section 6.

2. Formulation

A two-dimensional non-orthogonal coordinate system has two pairs of directions (Fig. 1). One pair, the covariant base vectors, describes the directions along the i-lines ($\partial \mathbf{r}/\partial j$) ($\mathbf{r}(i, j)$ is the position vector) and along the j-lines $\partial \mathbf{r}/\partial i$. On the other hand, the contravariant base-vectors ∇i and ∇j are perpendicular to the i-lines and j-lines. These directions are not independent but are related by the metric tensor which may itself be written in terms of either the covariant or contravariant base vectors. When writing equations involving the base vectors we may use the metric tensor to convert exclusively to either contravariant or covariant base vectors. Since $x(i, j)$, $y(i, j)$ and

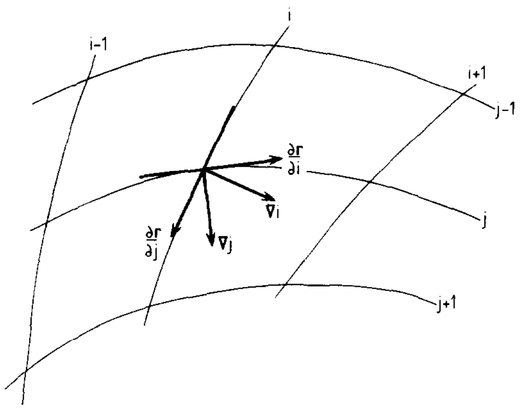

FIG. 1. A portion of the non-orthogonal (i, j) coordinate system showing the directions of the two types of base vectors.

not $i(x, y), j(x, y)$ are known—in the sense that a table of values for x and y against i and j exists—the covariant base vectors are naturally calculated:

$$\begin{aligned} \frac{\partial \mathbf{r}}{\partial i} &= \frac{\partial x}{\partial i}\hat{\mathbf{x}} + \frac{\partial y}{\partial i}\hat{\mathbf{y}}, \\ \frac{\partial \mathbf{r}}{\partial j} &= \frac{\partial x}{\partial j}\hat{\mathbf{x}} + \frac{\partial y}{\partial j}\hat{\mathbf{y}}, \end{aligned} \tag{1}$$

where $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ are unit vectors along the x and y axes.

The covariant metric tensor is normally expressed in terms of the covariant base vectors:

$$g_{ij} = \begin{pmatrix} \left(\frac{\partial \mathbf{r}}{\partial i}\right)^2 & \frac{\partial \mathbf{r}}{\partial i}\cdot\frac{\partial \mathbf{r}}{\partial j} \\ \frac{\partial \mathbf{r}}{\partial i}\cdot\frac{\partial \mathbf{r}}{\partial j} & \left(\frac{\partial \mathbf{r}}{\partial j}\right)^2 \end{pmatrix}.$$

The equations for the contravariant base vectors in terms of the covariant base vectors are

$$\begin{pmatrix} \nabla i \\ \nabla j \end{pmatrix} = \frac{1}{g} \begin{pmatrix} \left(\frac{\partial \mathbf{r}}{\partial j}\right)^2 & -\frac{\partial \mathbf{r}}{\partial i}\cdot\frac{\partial \mathbf{r}}{\partial j} \\ -\frac{\partial \mathbf{r}}{\partial i}\cdot\frac{\partial \mathbf{r}}{\partial j} & \left(\frac{\partial \mathbf{r}}{\partial i}\right)^2 \end{pmatrix} \begin{pmatrix} \frac{\partial \mathbf{r}}{\partial i} \\ \frac{\partial \mathbf{r}}{\partial i} \end{pmatrix}, \tag{2}$$

where g is the determinant of the metric tensor.

We require equations which describe a new set of lines which are perpendicular to the j-lines. This new set may either be represented by a tangent vector field $\mathbf{dr}(i, j)$ or by a function $k(i, j)$ which is constant along a new line. There are four equations for lines perpendicular to the j-lines (Fig. 2):

$$\mathbf{dr} \cdot \frac{\partial \mathbf{r}}{\partial i} = 0, \tag{3}$$

$$\mathbf{dr} \wedge \nabla j = 0, \tag{4}$$

$$\nabla k \cdot \nabla j = 0, \tag{5}$$

$$\nabla k \wedge \frac{\partial \mathbf{r}}{\partial i} = 0. \tag{6}$$

Substituting for **dr** and ∇k using

$$\mathbf{dr} = \frac{\partial \mathbf{r}}{\partial i} di + \frac{\partial \mathbf{r}}{\partial j} dj$$

$$\nabla k = \frac{\partial k}{\partial i} \nabla i + \frac{\partial k}{\partial j} \nabla j$$

and eliminating ∇i and ∇j using Eqs. (2); Eqs. (3) and (4) lead to

$$\frac{di}{dj} = f(i, j), \tag{7}$$

while Eqs. (5) and (6) lead to

$$\frac{\partial k}{\partial j} + f(i, j) \frac{\partial k}{\partial i} = 0, \tag{8}$$

where

$$f(i, j) = \frac{-\dfrac{\partial \mathbf{r}}{\partial i} \cdot \dfrac{\partial \mathbf{r}}{\partial j}}{\left(\dfrac{\partial \mathbf{r}}{\partial i}\right)^2},$$

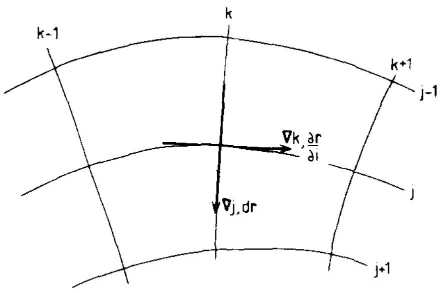

FIG. 2. A portion of the orthogonal (k, j) coordinate system illustrating equations (3)–(6).

which by Eqs. (1) is

$$f(i,j) = -\frac{\dfrac{\partial x}{\partial i}\dfrac{\partial x}{\partial j} + \dfrac{\partial y}{\partial i}\dfrac{\partial y}{\partial j}}{\left(\dfrac{\partial x}{\partial i}\right)^2 + \left(\dfrac{\partial y}{\partial i}\right)^2}. \tag{9}$$

Since di and dj describe $\mathbf{dr}$, the tangent to the orthogonal lines, Eq. (7) gives the rate of change of i with respect to j along an orthogonal trajectory.

Equation (8) is almost the simplest non-trivial first order partial differential equation. Being first order, it is hyperbolic and has one set of real characteristic directions or characteristics (see for example Chester [9]). With equations of this type an equivalent system of ordinary differential equations, describing the rate of change of k, i and j along the characteristics, may be derived. For Eq. (8), this system is

$$di/dt = f(i,j),$$
$$dj/dt = 1,$$
$$dk/dt = 0,$$

where t is a running parameter along the characteristics. These equations are just an expanded form of Eq. (7) and we may deduce that the characteristics of Eq. (8) are the orthogonal trajectories.

3. Difference Form of the Orthogonal Equations

Either of the equivalent forms (Eqs. (7), (8)) which define the coordinate orthogonal to j may in principle be used. Since the ordinary differential Eq. (7) gives the rate of change of i with j along an orthogonal trajectory, which is unknown a priori, a second order scheme may only be used here at the expense of iteration. In contrast, however, the partial differential Eq. (8) readily admits the use of a scheme of second order accuracy.

The difference scheme chosen for Eq. (8) is implicit and of second order accuracy and is illustrated by Fig. 3. For simplicity i and j are assumed to have unit intervals between adjacent points. The scheme is centred about the point $(i, j+\frac{1}{2})$; the representation is as follows:

$$(\partial k/\partial j)(i, j+\tfrac{1}{2}) = k(i, j+1) - k(i,j), \tag{10}$$

$$(\partial k/\partial i)(i, j+\tfrac{1}{2}) = \tfrac{1}{4}(k(i+1, j+1) - k(i-1, j+1) + k(i+1, j) - k(i-1, j)), \tag{11}$$

$$f = f(i, j+\tfrac{1}{2}). \tag{12}$$

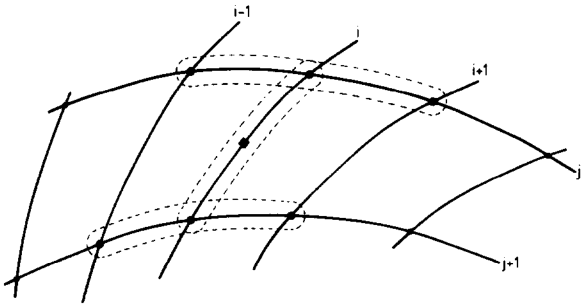

FIG. 3. Two j-lines illustrating the finite difference scheme.

Substituting Eqs. (10)–(12) into Eq. (8) gives

$$-\frac{f}{4}k(i-1,j+1)+k(i,j+1)+\frac{f}{4}k(i+1,j+1)$$

$$=\frac{f}{4}k(i-1,j)+k(i,j)-\frac{f}{4}k(i+1,j), \qquad i=2,3,...,I-1. \tag{13}$$

Equations (13) are an incomplete set of equations for $k(i,j+1)$, $i=1,2,...,I$, in terms of $k(i,j)$, $i=1,2,...,I$, or vice versa. They must be supplemented by two equations, one for each end. These two equations depend on the nature of the coordinate system: there are three cases.

If the j-lines are closed, then, in principle, there are no ends. In practice we have to have ends but, nevertheless, the same difference scheme may be used at each point. Although k is a single-valued function of i, it is a many-valued function of position. ∇k must be single-valued and, for this to be so, the different branches of k must differ by an integer times a constant period Δk, where

$$\Delta k = k(I,j)-k(1,j)$$

and the $i=1$ line, the $i=I$ line and the branch cut coincide. Rather than define $k(I,j)$ we may define Δk and use

$$\begin{aligned} k(0,j)&=k(I-1,j)-\Delta k\\ k(I,j)&=k(1,j)+\Delta k \end{aligned} \tag{14}$$

to write Eqs. (13) for $i=1,2,...,I-1$. The matrix of the coefficients of $k(i,j+1)$ in this case is not quite tridiagonal [10]; it has two extra non-zero elements: in the top right-hand corner and in the bottom left-hand corner. Even so, the system may easily be solved by Gaussian elimination and back substitution.

Although the j-lines are not closed, the case of translational symmetry (as in a crystal lattice) is entirely analogous and deserves no separate explanation.

If the j-lines are open, Eq. (11) may be replaced by one-sided difference equations at the ends, for example,

$$\begin{aligned}(\partial k/\partial i)(1,j) &= k(2,j) - k(1,j),\\ (\partial k/\partial i)(I,j) &= k(I,j) - k(I-1,j).\end{aligned} \tag{15}$$

In this case the system of equations is tridiagonal which is particularly quick and easy to solve [10]. For higher accuracy, but at the expense of a more difficult system of equations to solve, a second order, three point scheme may be used.

In the original non-orthogonal coordinate system, the end i-lines are often already orthogonal. In this case Eqs. (13) may be supplemented by

$$\begin{aligned}k(1,j+1) &= k(1,j),\\ k(I,j+1) &= k(I,j).\end{aligned} \tag{16}$$

The system of equations will then be tridiagonal.

It seems natural to regard the two additional equations at $i=1$ and $i=I$ ($i=I-1$ in the periodic case) as boundary conditions. This is proper in the third case (Eqs. (16)) which, in general, overspecifies the problem; but in the particular case when the i-lines are already orthogonal, the over-specification and Eq. (8) are consistent. However, in the other two cases the equations at the ends are not boundary conditions—they are difference equations representing Eq. (8).

4. Production of the Orthogonal Mesh

Given the values of k on a j-line we are now able to calculate k on an adjacent j-line. From the initial conditions of k on a single j-line, we may, therefore, calculate k at every point on the mesh. A convenient initial condition is

$$k(i,1) = i, \qquad i = 1, 2, \ldots, I;$$

$k(i,2)$ is found by solving the system of Eqs. (13). Then, putting $k(i,2)$ on the right-hand side of Eqs. (13), we may solve for $k(i,3)$ and so on, until we have $k(i,j)$ for i in the range 1 to I and j in the range 1 to J. Since we have x, y and k at each point we may regard x and y as functions of k and j. The cartesian coordinates on an i-line must now be changed such that k is a constant on the new lines: the k-lines. Holding j constant, we may find the coordinates x, y at the values $k = 1, ,2, \ldots, I$ by interpolation. Choosing these values preserves the points on the first j-line (Fig. 4).

In practice some modifications are required to treat the occurrence of focussing and defocussing of lines leading to irregular intervals in k—something highly detrimental to accurate interpolation. Rather than calculate k at each point it is

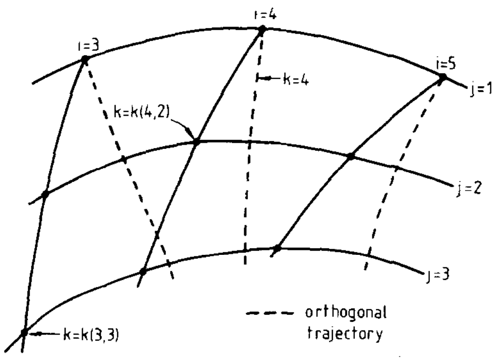

FIG. 4. The function $k(i, j)$ with the initial conditions of k equal to i on the first j-line.

preferable to calculate i^* at each point, where i^* is the "i" of Eq. (7). Let the initial conditions for i^* be

$$i^*(i, 1) = i, \qquad i = 1, 2, \ldots, I. \tag{17}$$

The orthogonal trajectory, starting at $(i, 1)$ cuts the jth line with an i-value of $i^*(i, j)$ (Fig. 5).

Say that $i^*(i, j)$ has been found and that $i^*(i, j+1)$ is required. We find the values of k on the jth line which are consistent with k being equal to i on the $(j+1)$th line. That is, writing

$$k(i, j+1) = i, \qquad i = 1, 2, \ldots, I.$$

We swap the right- and left-hand sides of Eqs. (13) and solve for $k(i, j)$. Having found k at the non-orthogonal points on the jth line we then find, by interpolation, the values of k at the orthogonal points (where i is equal to i^*). The intervals in i are

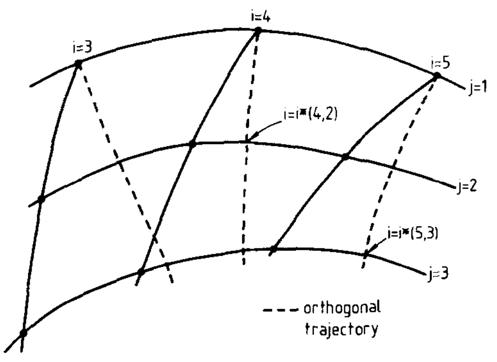

FIG. 5. The function $i^*(i, j)$ with the initial conditions of i^* equal to i on the first j-line.

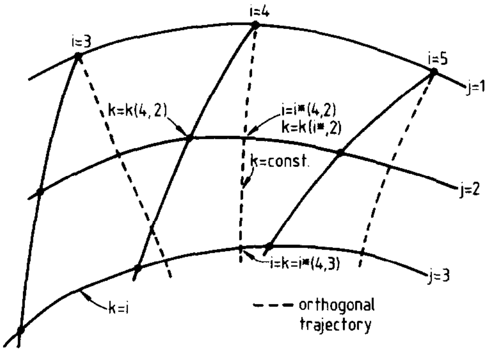

FIG. 6. Illustrating how $i^*(i, j)$ is obtained from $k(i, 2)$ and $i^*(i, 1)$. The initial conditions are that $k(i, 3)$ is equal to i.

equal which makes the interpolation simple, quick and accurate. Four point Lagrange interpolation was used, for example, if $i^*(i, j)$ lies in the range i' to $i' + 1$:

$$\begin{aligned} k(i^*, j) = &-\frac{p(p-1)(p-2)}{6} k(i'-1, j) + \frac{(p^2-1)(p-2)}{2} k(i', j) \\ &- \frac{p(p+1)(p-2)}{2} k(i'+1, j) + \frac{p(p^2-1)}{6} k(i'+2, j), \end{aligned} \tag{18}$$

where $p = i^* - i'$.

Since k is a constant on orthogonal trajectories and since k is equal to i on the $(j+1)$th line (Fig. 6):

$$i^*(i, j+1) = k(i^*(i, j), j) \tag{19}$$

Starting from the initial conditions (17) we may use the above to calculate i^* at each point. $i^*(i, j)$ is the ideal information for finding the cartesian coordinates of the orthogonal points; it is the value of i to which the (i, j)th point must be moved. The interpolation is identical to that of Eq. (18).

5. Summary of the Orthogonalization Procedure

To clarify and stress the procedure the major steps to be executed are outlined below:

(a) The function $f(i, j)$ (Eq. (9)) is calculated on the non-orthogonal mesh. In practice, Eqs. (10) and (11) were used to calculate the derivatives of x and y in Eq. (9).

(b) i^* is set to its initial value on the first j-line (Eq. 17).

(c) The system of Eqs. (13) is solved backward for $k(i, 1)$ with $k(i, 2)$ equal to i. That is, the following system of equations is solved with j equal to 1:

$$\frac{f}{4}k(i-1,j) + k(i,j) - \frac{f}{4}k(i+1,j) = \frac{f}{2} + i.$$

(d) $i^*(i, 2)$ is found by interpolation of $k(i, 1)$ (Eqs. (18), (19) with j equal to 1).

(e) Steps (c) and (d) are repeated for the second and third j-lines; and then the third and fourth j-lines; and so on until i^* is known at every point.

(f) The points are moved to their orthogonal positions by practising the interpolation of Eq. (18) on x and y.

6. Examples

To avoid the occurrence of larger area elements than necessary, the initial conditions of an even distribution of i-points should be taken on a j-line passing through the most divergent region of the i-lines. For closed j-lines, this is achieved by starting at the outside j-line and working in. This was applied in Figs. 7 and 8 and consequently the outside points of the orthogonal and nonorthogonal meshes coincide.

The procedure cannot cope with an unlimited amount of shear as can the method described in [8]. When there is shear the distances between adjacent points on an i-line may be large and, even if $f(x, y)$ is smooth, $f(i, j)$ may not be. If the distances become unacceptably large more j-lines may be inserted. However Fig. 7 shows that the method deals satisfactorily with a high shear; the circles are also shifted.

The non-orthogonal mesh of Fig. 8 is an example of a method of producing a smooth non-orthogonal mesh from the innermost and outermost j-lines. The mesh was constructed by joining the points of the innermost and outermost j-lines by

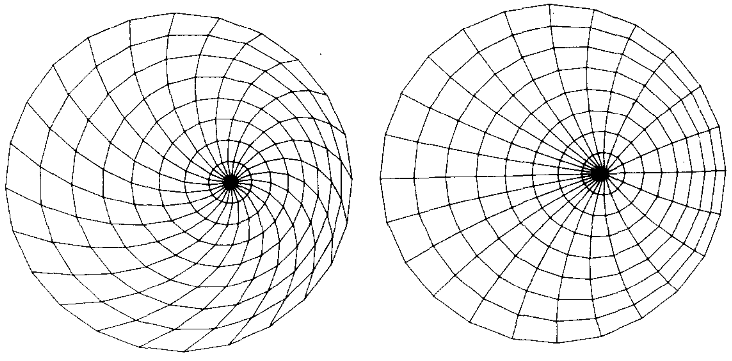
FIG. 7. Non-concentric and sheared circles.

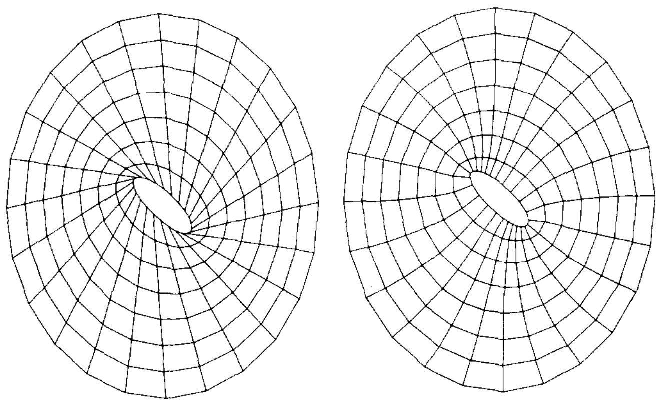

FIG. 8. The non-orthogonal coordinate system was constructed by joining the points on the inner and outer ellipses by straight lines.

straight lines and placing interior points at equal intervals along the straight lines. The non-orthogonal meshes of Figs. 9 and 10 were constructed in the same way. This is a simple and general method of producing interior j-lines which deform smoothly from the outside shape to the inside shape.

In Figs. 9 and 10 the j-lines are horizontal and the initial conditions were taken on the bottom boundary. In these two examples either the periodic treatment (Eqs. (14)) or the initial boundary value treatment (Eqs. (16)) could have been used; the former was used in Fig. 9 and the latter in Fig. 10. Figure 10 shows enhancement of the point density in the central region by a judicious choice of the initial conditions.

The quality of the orthogonal mesh does depend upon the choice of the non-orthogonal mesh. The difference between orthogonal meshes generated from different nonorthogonal systems having the same j-lines and boundary points is a measure of error in the procedure. Figure 11 makes such a comparison and the error shows as a slight thickening of some of the i-lines where the two orthogonalized meshes do not quite coincide. Even if we accept that the scheme produces the same mesh from all non-orthogonal meshes having the same j-lines and boundary points, different choices of j-lines will produce different orthogonal meshes, and some will be better than

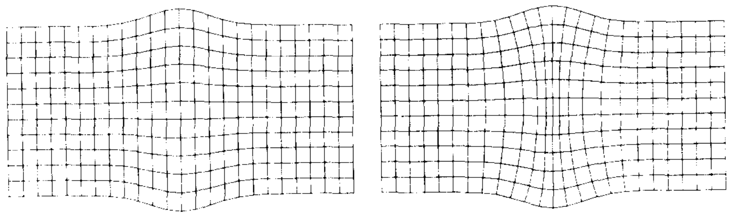

FIG. 9. The j-lines are open and run horizontally. In this example, periodic conditions were applied at the ends.

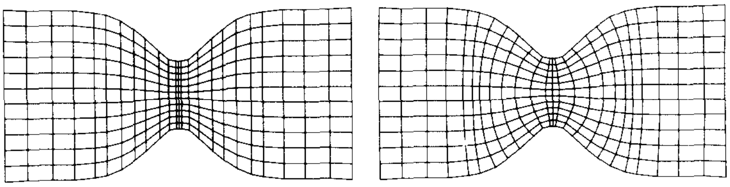

FIG. 10. The j-lines are open and run horizontally. In this example, overspecified boundary conditions were applied at the ends.

others. Apart from choosing j-lines which are reasonably spaced and smooth, the procedure offers no guidance in choosing j-lines which optimize the orthogonalized mesh. Figure 12 demonstrates that the procedure does not rely on the j-lines being equally spaced.

An orthogonalization procedure using Eqs. (15) has not been written and tested. It was felt that this case is less useful because the boundary of the mesh is changed by the procedure. In practice it has always been possible to construct open j-lines which cut the end i-lines at right angles as in Figs. 9 and 10.

It is very difficult to make quantitative judgement of the orthogonality of the orthogonal mesh because the angle of intersection of an i-line and a j-line depends on the interpolation used to construct a line from a few points. Clearly, as the point density is increased, Eqs. (10) and (11) become better approximations to the continuous derivatives, and the resulting mesh is more accurately orthogonal. In the examples of Figs. 7–10 the point densities are rather low but the procedure still produces meshes which appear tolerably orthogonal.

One source of difficulty in obtaining a quantitative measure of the error of an orthogonalized mesh is the absence of a discrete definition of an accurately orthogonal mesh. Following the argument in the introduction one could use the deviation of $\nabla i \cdot \nabla j$ from zero as a measure of non-orthogonality. However, as pointed out at the beginning of Section 2, ∇i and ∇j are poorly represented when i and j are considered to be functions of x and y and $\nabla i \cdot \nabla j$ is identically zero when i and j are considered to be functions of i and j. This problem may be overcome by

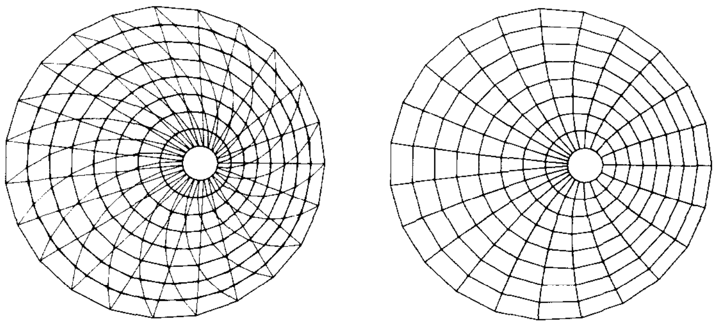

FIG. 11. A comparison of two different orthogonal systems having the same j-lines and boundary points. The different orthogonal systems are superimposed on the right.

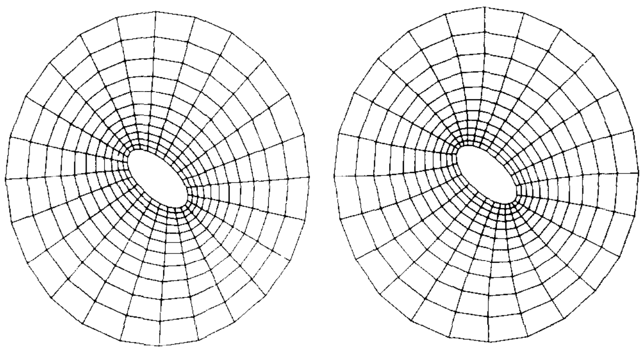

FIG. 12. Varying j-line spacing.

calculating the product $(\partial \mathbf{r}/\partial i) \cdot (\partial \mathbf{r}/\partial j)$ using Eqs. (1). This measure implicitly assumes that the "accurate" discrete mesh is that for which a finite difference approximation of $(\partial \mathbf{r}/\partial i) \cdot (\partial \mathbf{r}/\partial j)$ is smallest. This assumption may be avoided by measuring the error in calculating a known invariant. The invariant chosen here is

$$e = \nabla x \cdot \nabla y,$$

where e should accurately be zero on any coordinate system Considering e, x and y to be functions of i and j

$$e(i,j) = \frac{\partial x}{\partial i}\frac{\partial y}{\partial i}(\nabla i)^2 + \frac{\partial x}{\partial j}\frac{\partial y}{\partial j}(\nabla j)^2 + \left(\frac{\partial x}{\partial i}\frac{\partial y}{\partial j} + \frac{\partial y}{\partial i}\frac{\partial x}{\partial j}\right)\nabla i \cdot \nabla j, \tag{20}$$

but if i and j are orthogonal

$$e(i,j) = \frac{\partial x}{\partial i}\frac{\partial y}{\partial i}(\nabla i)^2 + \frac{\partial x}{\partial j}\frac{\partial y}{\partial j}(\nabla j)^2. \tag{21}$$

A finite difference approximation for $e(i,j)$ was calculated on each of the orthogonalized meshes in Figs. 7–10, based on Eq. (21). In general, the result differs from zero for two reasons: error associated with the finite difference approximation in (i,j) space and error associated with non-orthogonality. If the mesh is non-orthogonal then the absence of the last term in Eq. (20) will cause e to deviate from its true value by an amount proportional to $\nabla i \cdot \nabla j$. This deviation should be compared with unity—the magnitude of ∇x and ∇y. In all cases the maximum deviation was less than 0.025, with root mean square deviations of between 0.002 (Fig. 9) and 0.01 (Fig. 8). These figures suggest that non-orthogonality error and finite difference error are comparable.

The time taken by the procedure depends on whether the periodic treatment or the initial boundary value treatment is being applied, the former taking longer. The time taken to solve the system of equations (13), unlike more complicated systems, scales linearly with the number of equations—or with the number of i-points on each j-line. The system of equations (13) must be solved $(J-1)$ times and since all other operations depend on the number of mesh points the time taken by the procedure scales linearly with the number of mesh points. The program took between 0.6 and 0.75 sec per 1000 points on a CDC 6500 (about three times slower than a CDC 6600). Some improvement on these times may be possible as no attempt has been made to optimise the speed of the program.

7. Discussion

An orthogonalization procedure which deals with open and closed lines has been described. The procedure is an addition to the list [2 – 8] of methods already available for the production of boundary fitted orthogonal coordinates, and an alternative to [8] (which also preserves j-lines) for use every time step in a "waterbag" code.

Two advantages of the procedure are its speed and simplicity, (about 100 FORTRAN statements). The procedure should not be considered an exotic method which is only justified in the solution of large problems on large computers. However, if more time is available for the production of the mesh, greater accuracy may be obtained by using a larger number of points in the orthogonalization than in the subsequent numerical solution of the differential equations.

The essence of the orthogonalization procedure is Eq. (8). The numerical procedure described in Sections 3 and 4 is meant as a suggestion; it is unlikely that this procedure is the best (in terms of speed and accuracy) in all circumstances. However, the numerical procedure does show that it is feasible to employ a finite difference representation of Eq. (8) to produce discrete orthogonal coordinate systems.

General three-dimensional orthogonal coordinate systems are considerably more complex than two-dimensional coordinates. Darboux's theorem [12] implies that, given one set of coordinate surfaces, then another two sets (required to complete an orthogonal triad) may not exist. If an orthogonal system is to be produced from a non-orthogonal system then, is general, none of the original coordinates may be conserved. Since the procedure described here is dependent on a conserved coordinate, it seems unlikely that the procedure can be directly applied to the production of discrete coordinates in three dimensions.

References

1. P. M. Morse and H. Feshbach, "Methods of Theoretical Physics," Part I, p. 115, McGraw–Hill, New York, 1953.

2. S. K. Godunov and G. P. Prokopov, *USSR Comp. Math. Phys.* **7** (1967), 89.
3. W. D. Barfield, *J. Computational Phys.* **5** (1969), 23.
4. W. D. Barfield, *J. Computational Phys.* **6** (1970), 417.
5. R. Meyder, *J. Computational Phys.* **17** (1975), 53.
6. T. K. Hung and T. D. Brown, *J. Computational Phys.* **23** (1977), 343.
7. G. Starius, *Numerische Mathematik.* **28** (1977), 25.
8. D. E. Potter and G. H. Tuttle, *J. Computational Phys.* **13** (1973), 483.
9. C. R. Chester, "Techniques in Partial Differential Equations," McGraw–Hill, New York, 1971.
10. D. E. Potter, "Computational Physics," Wiley, New York/London, 1973.
11. D. E. Potter, *in* "Methods in Computational Physics" (B. Alder, S. Fernbach, and M. Rotenburg, Eds.), Vol. 16, p. 43, Academic Press, New York, 1970.
12. C. E. Weatherburn, "Differential Geometry of Three Dimensions," Art. 112, Cambridge Univ. Press, London, 1927.