MATHEMATICS OF COMPUTATION, VOLUME 33, NUMBER 147
JULY 1979, PAGES 953–969

Numerical Conformal Mapping

By Sukumar Chakravarthy and Dale Anderson

Abstract. A numerical procedure to determine the discrete conformal mapping of an arbitrary simply connected region onto the open unit disk is described. The method is fast and directly provides an estimate of the global error due to the discretization of the mapping. The intimate relationship between the Riemann Mapping Theorem and the numerical method of construction of the conformal mapping is brought out in the presentation. The procedure is an interesting example of a method of construction that is directly based on a theorem guaranteeing existence and uniqueness.

1. Introduction.

Coordinate transformations are very commonly employed in various fields to map complex geometries into simpler domains. Conformal transformations preserve the angle between the tangents to any two curves at their point of intersection. Thus, an orthogonal coordinate system in one plane would correspond to orthogonal lines in the image plane. This property is very desirable in applications such as flow field computations using finite difference methods [8]. Boundary conditions are easily and accurately applied and experience also indicates that more accurate solutions are obtained with orthogonal coordinate systems.

As early as 1931, Theodorsen had developed a series method to conformally map the exterior of airfoils of finite thickness into the exterior of the unit circle [1, pp. 50–60]. Moretti [8] has described a technique to obtain conformal grids about complex geometries such as airframes but part of the boundary evolves from the transformation and cannot be fixed a priori in his method. Barfield [2] has succeeded in developing a general numerical method to compute the conformal transformation between any two-dimensional simply connected region and the unit circle. A new, equally general method is described in this paper. In the concluding remarks, the methods listed above are compared briefly and the advantages and disadvantages of the new method are pointed out. A distinguishing feature of the method described is that it logically evolves from the existence and uniqueness theorem for conformal mappings. Such a formulation preserves the appeal and generality of the method regardless of possible applications.

The Riemann Mapping Theorem [4, p. 156] asserts the existence and uniqueness (up to a rotation) of a conformal mapping between any simply connected domain and the unit disk, once the point that is to be mapped into the origin of the unit disk has been chosen. Let us call the simply connected domain the physical domain and the

Received May 24, 1978.
AMS (MOS) subject classifications (1970). Primary 30A28.
Key words and phrases. Conformal mapping, numerical conformal mapping.

unit disk the computational domain. In the discrete sense, the Riemann Mapping Theorem implies that for a conformal map, there exists a unique one-one correspondence between chosen points in the computational domain and their image points in the physical domain. In other words, for an arbitrary choice of discrete points in the physical domain chosen to correspond to selected points in the computational domain, a conformal map does not exist. The aim of the numerical conformal mapping is to find the set of points in the physical domain that is conformally mapped into the selected set of points in the computational domain.

The physical and computational domains are depicted in Figure 1. A conformal mapping is governed by the Cauchy-Riemann equations which can be written in polar coordinates as

$$\frac{\partial X}{\partial r} = \frac{1}{r}\frac{\partial Y}{\partial \theta}, \quad \frac{1}{r}\frac{\partial X}{\partial \theta} = -\frac{\partial Y}{\partial r}. \tag{1.1}$$

A discrete or numerical conformal mapping is governed by the difference analogues of the Cauchy-Riemann equations (C-R equations) which we shall call the discrete C-R equations. The discrete C-R equations are obtained by employing the finite difference method on uniformly spaced mesh points in the computational domain.

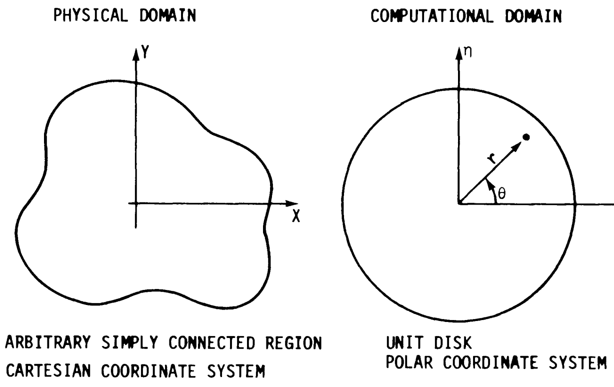

FIGURE 1
Physical and computational domains

The numerical conformal mapping proceeds iteratively. A trial boundary point distribution is chosen in the physical plane. This is quantified by a set of (X, Y) pairs. It is possible to use only the Y values and compute the corresponding set of X'' values at the boundary points that would satisfy the discrete C-R equations in the computational domain. The difference between the X and X'' values constitutes a residue. The Riemann Mapping Theorem assures us that this residue would in general be nonzero for the trial distribution of points. Parameter optimization techniques are used to change the boundary point distribution in a such a manner as to drive the residue to zero. The distribution of boundary points corresponding to a vanishing residue is the correct, required set of boundary points. Using this boundary data, the discrete C-R equations are solved to give the discrete conformal mapping.

2. The Discrete C-R Equations and Boundary Conditions. Consider the equations

$$r\frac{\partial V}{\partial r}+\frac{\partial U}{\partial \theta}=0, \quad \frac{\partial V}{\partial \theta}-r\frac{\partial U}{\partial r}=0. \tag{2.1}$$

The C-R equations given by (1.1) are equivalent to the equations above if one of the following associations are made:

$$V=Y, \quad U=X; \quad V=X, \quad U=-Y.$$

The former association conforms to the roles of X and Y described in the outline of the method given in the introduction. The latter association lets the roles be reversed. For generality, let us proceed in terms of Eqs. (2.1).

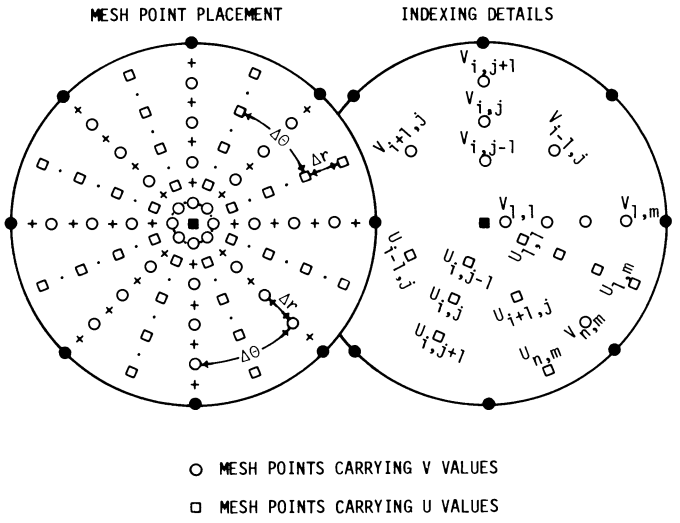

FIGURE 2
Finite difference mesh

Figure 2 portrays the mesh in the circular computational domain with the unit circle shown as a full line. The U data are carried over the mesh points denoted by the squares and the V data carried over the mesh points denoted by the circles. The i and j indices denote the θ and r locations for the U and V mesh points. The placement of mesh points is shown on the left half of Figure 2 and the indexing is explained on the right half. For this staggered grid the discrete C-R equations that we will consider are given by the second order accurate finite difference formulas shown below.

$$r'_{i,j}\frac{V_{i,j+1}-V_{i,j}}{\Delta r}+\frac{U_{i+1,j}-U_{i,j}}{\Delta\theta}=0, \tag{2.2a}$$

$$\frac{V_{i,j}-V_{i-1,j}}{\Delta\theta}-r''_{i,j}\frac{U_{i,j}-U_{i,j-1}}{\Delta r}=0. \tag{2.2b}$$

Here $r'_{i,j}$ corresponds to the radius of the position denoted by the cross (+), where (2.2a) is balanced and $r''_{i,j}$ corresponds to the radius of the position denoted by the dot ($\cdot$), where (2.2b) is balanced. The discretization described in this section and the solution procedure described in the next to solve the discrete C-R equations in polar coordinates follow from Lomax and Martin's work on the C-R equations in Cartesian coordinates [7].

TABLE 1
System of discrete C-R equations

$$\left[\begin{array}{ccccc|ccccc}
A & & & & & -I & I & & & \\
 & A & & & & & -I & I & & \\
 & & \ddots & & & & & \ddots & \ddots & \\
 & & & A & & & & & -I & I \\
 & & & & A & I & & & & -I \\
\hline
I & & & & -I & B & & & & \\
-I & I & & & & & B & & & \\
 & \ddots & \ddots & & & & & \ddots & & \\
 & & -I & I & & & & & B & \\
 & & & -I & I & & & & & B
\end{array}\right]
\left[\begin{array}{c}
V_1 \\ V_2 \\ \vdots \\ V_{n-1} \\ V_n \\ \hline U_1 \\ U_2 \\ \vdots \\ U_{n-1} \\ U_n
\end{array}\right]
=
\left[\begin{array}{c}
f_1 \\ f_2 \\ \vdots \\ f_{n-1} \\ f_n \\ \hline g_1 \\ g_2 \\ \vdots \\ g_{n-1} \\ g_n
\end{array}\right]$$

$$B = \begin{bmatrix}
-\mu'_1 & & & & \\
\mu'_2 & -\mu'_2 & & & \\
 & \mu'_3 & -\mu'_3 & & \\
 & & \ddots & \ddots & \\
 & & & \mu'_n & -\mu'_n
\end{bmatrix}
\qquad
A = \begin{bmatrix}
-\mu''_1 & \mu''_1 & & & \\
 & -\mu''_2 & \mu''_2 & & \\
 & & -\mu''_3 & \mu''_3 & \\
 & & & \ddots & \ddots \\
 & & & & -\mu''_n
\end{bmatrix}$$

$$\mu = \frac{\Delta\theta}{\Delta r} \qquad \begin{array}{l} \mu' = \mu\, r'_{i,j} \\ \mu'' = \mu\, r''_{i,j} \end{array} \quad \text{for all } i$$

The notation V_i is used to denote the vector $\begin{bmatrix} V_{i,1} \\ V_{i,2} \\ \vdots \\ V_{i,m} \end{bmatrix}$

The same notation is implied in U_i, f_i & g_i,
$f_{i,j} = 0$ and $g_{i,j} = 0$ for $i = 1, \dots, n; j = 1, \dots, m$
except
$f_{i,m} = -\mu'_{i,m} V_{i,m}$ and $g_{i,1} = -r''_{i,1} U_0$ for $i = 1, \dots, n$.

Collecting all the difference equations obtained at mesh points denoted by the U and V indices $i = 1, \ldots, n$, and $j = 1, \ldots, m$, a system of $2 \times m \times n$ difference equations is obtained for the unknown U and V values. This is displayed as a matrix system of equations in Table 1. From the right-hand side of the equations it is evident that the $m \times n$ U values and $m \times n$ V values can be determined if the n-boundary values of V on the unit circle and the value of U at the center of the unit disk are pre scribed. These are the boundary conditions. Shaded circles and squares are employed in Figure 1 to signify points where boundary data are needed.

3. Solving the Discrete C-R Equations. The difference equations given in Table 1 couple the U and V values. Let the required boundary values of U and V be given, and let it be required to solve the difference equations to obtain the U and V values at the mesh points. The first step in the solution procedure is to decouple the U and V values.

1. The second row in the bottom half of Table 1 is subtracted from the first and B times the first row in the top half is added to the result to give the first decoupled equation for V.

2. The third row in the bottom half of Table 1 is subtracted from the second and B times the second row in the top half is added to the result to give the second decoupled equation for V.

3. This is repeated for all rows in the bottom half except the last. The first row in the bottom half is subtracted from the last and B times the last row in the top half is added to the result to give the last decoupled equation for V.

This procedure results in the following set of equations for the V values uncoupled from the U values.

$$\begin{bmatrix} BA+2I & -I & & -I \\ -I & BA+2I & -I & \\ & -I & BA+2I & -I \\ & \ddots & \ddots & \ddots \\ -I & & -I & BA+2I \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ \vdots \\ V_n \end{bmatrix} = \begin{bmatrix} g_1 - g_2 + Bf_1 \\ g_2 - g_3 + Bf_2 \\ g_3 - g_4 + Bf_3 \\ \vdots \\ g_n - g_1 + Bf_n \end{bmatrix}. \tag{3.1}$$

The coefficient matrix is of the periodic block tridiagonal form with all the diagonal block matrices being the same matrix $C = BA + 2I$. The matrices A and B are upper and lower bidiagonal, respectively. Hence the matrix C is tridiagonal. Thus, the system of equations (3.1) can be solved very rapidly by direct cyclic reduction methods such as that detailed by Sweet [10]. Once the V values are known, the U values can be computed very easily. For example, the equations in the bottom half of Table 1 could be used for that purpose.

It is of interest to note that the decoupling procedure is the discrete equivalent of differentiating the second of Eqs. (2.1) with respect to θ, differentiating the first with respect to r and adding the resulting two equations to yield a Laplace equation for V in polar coordinates. In addition, the decoupling procedure implicitly imposes the boundary conditions. Thus, it happens that the decoupled system of equations corresponds to a finite difference approximation to the Laplace equation for V with a Dirichlet boundary condition on the outer boundary (i.e. the unit circle), a Neumann boundary condition near the center of the disk, and a periodic boundary condition in the circumferential direction. This becomes apparent when the matrix BA is written out in full detail.

4. Formulation of the Numerical Conformal Mapping. In the previous two sections a procedure was presented to determine the unknown U and V values at the discrete mesh points from the prescribed V values on the unit circle and the prescribed U values at the center of the disk. The U and V values are carried over different sets of mesh points. However, if the bounding curve of the physical domain is smooth, one could use interpolation formulas to determine from the solution, the U values at the boundary mesh points where V values are prescribed. Thus, when the discrete C-R equations are satisfied in the interior, it is possible to use the V values prescribed at mesh points on the unit disk to compute the U values at the same points. The V value at the center of the unit disk where the U value is prescribed could also be computed by interpolation.

It is appropriate to note that the requirement of smoothness of the curve bounding the physical domain is not a restriction. Any sharp corners in the bounding curve could be removed by employing scaled Karman-Trefftz transformations. This has been used by Moretti [8]. On successive applications of the scaled Karman-Trefftz transformations, the physical domain with corners in its bounding curve is transformed into an intermediary domain with a smooth bounding curve to which the numerical conformal mapping procedure can be applied. Needless to say, the result of all the successive mappings is a mapping, conformal everywhere except at the corners of the bounding curve of the physical domain.

Before proceeding any further let us observe some consequences of solving the linear algebraic system of discretized C-R equations. It is clear upon close inspection of the system that if the prescribed V values are all zero and the prescribed U value is U_0, then the solution is given by

$$U_{i,j} = U_0 \quad \text{and} \quad V_{i,j} = 0 \quad \text{for } i = 1, \ldots, n; j = 1, \ldots, m.$$

Therefore, the computed boundary U values would all be U_0. Let us construct an influence matrix A as follows: The ith column of A consists of the computed boundary U values when the prescribed U_0 value is zero and all the prescribed V values are zero except for the ith which is set to unity. Each of these solutions is known as a fundamental solution. Let β'' be the vector made up of the computed boundary U values, when the vector of prescribed V values is denoted by γ and the prescribed U value is U_0. Clearly,

$$\beta'' = A\gamma + U_0. \tag{4.1}$$

Here and in the subsequent equations, when the scalar U_0 is shown being added to a vector, it is intended for U_0 to be added to all elements of that vector.

This sets the stage for formulating the numerical conformal mapping. Let the boundary of the physical domain be described parametrically in terms of the running parameter α (for example the length along the arc of the bounding curve) by

$$U = \beta(\alpha), \qquad V = \gamma(\alpha). \tag{4.2}$$

Let us pick discrete α_i, $i = 1, \ldots, n$, to identify the points that are to be mapped conformally into the uniformly spaced boundary mesh points. To each α_i there corresponds a β_i and a γ_i. Let us also pick U_0. The equation (4.1) gives the boundary U values corresponding to the γ and U_0 values, which are compatible with the discrete C-R equations being satisfied in the interior. The values β'' should be equal to the values β but will not be for wrong choices of the values α. Let us then compute the residue

$$R = \beta'' - \beta = A\gamma + U_0 - \beta. \tag{4.3}$$

The object of the numerical conformal mapping is to choose the α_i, $i = 1, \ldots, n$, in such a way as to force the residue to zero.

The Riemann Mapping Theorem shows us that we have to fix the point that is to be mapped into the origin of the unit disk and the angular orientation to guarantee a unique conformal mapping. It has been shown in this section that U_0 is specified naturally as boundary data. The angular orientation is also easily fixed, for example by stipulating that α_n be fixed at a chosen value (fixing the point that is mapped into the origin and the point on the physical domain boundary that is mapped into a chosen point on the unit circle fixes the angular orientation too). It will be shown in the next section how to fix V_0, the V value of the point that is mapped into the origin. As a prerequisite, we compute for each fundamental solution, the interpolated V value at the center of the unit disk. Let us store these values in the vector δ. Therefore, one way of fixing V_0 is by choosing the α_i, $i = 1, \ldots, n - 1$, to satisfy

$$\delta^T\gamma - V_0 = 0 \tag{4.4}$$

in addition to forcing the residue to zero.

5. **Minimizing the Residue.**

The basic numerical conformal mapping problem has been formulated in the last section in terms of solving a system of nonlinear algebraic equations given by

$$R = A\gamma(\alpha) + U_0 - \beta(\alpha) = 0. \tag{5.1}$$

In this section it is shown how to employ parameter optimization methods to minimize the residue in an attempt to force the residue to zero. It is also shown how the penalty function method can be used to fix V_0. The parameter optimization techniques are only one choice of methods to solve (5.1). Other methods for solving (5.1) are equally acceptable if they possess equal speed of computation.

In order to apply minimization techniques we first construct the preliminary loss function f' given by

$$f' = R^TR = \tfrac{1}{2}(A\gamma + U_0 - \beta)^T(A\gamma + U_0 - \beta). \tag{5.2}$$

The absolute minimum that f' can achieve is zero and at that point, each element of the residue vector R is also zero. It is also required to enforce our choice of V_0. To do this within the framework of parameter optimization methods, we employ the penalty function technique [5, pp. 53–59]. A penalty function measuring the degree of dissatisfaction of (4.4) is appended to the preliminary loss function f' to yield another loss function f.

(5.3) $$f = f' + \tfrac{1}{2} K_\nu(\delta^T\gamma - V_0)^2.$$

The problem of minimizing f' subject to the constraint that $\delta^T\gamma$ be equal to V_0 is replaced by a sequence of unconstrained subproblems of minimizing f. The subscript ν denotes the νth subproblem. The penalty constant K_ν is increased between every subproblem; and in practice, after a few subproblems, the required minimum satisfying the constraint is obtained. The convergence of this method has been proved in the reference cited.

Two methods to minimize f for each subproblem are detailed below. They are the Conjugate Gradient method (CG1) explained in [6] and a modified Newton method. The basic Newton method can be found in [9]. The loss function f depends upon the values of the parameters α_i. Parameter optimization methods find the optimum values for the parameters α_i that minimize f. Both methods detailed have the common scheme

(5.4) $$\begin{aligned} &\alpha^0 = \text{arbitrary},\\ &\alpha_i^{k+1} = \alpha_i^k + \phi^k p_i^k, \quad i = 1, \dots, n-1; k = 0, 1, \dots,\\ &\phi = \text{positive}. \end{aligned}$$

The superscript k is the iteration number. The vector p defines the direction of search. The scalar ϕ is the optimum step size in that direction of search and is determined by a one-dimensional or linear search procedure such as that described in [6].

The directions of search are chosen differently for the two minimization techniques. For the CG1 method

(5.5) $$\begin{aligned} &p^0 = -g^0, \quad \text{(steepest descent direction)},\\ &p^k = -g^k + \mu^k p^{k-1}, \qquad k = 1, 2, \dots, \quad \text{(conjugate directions)},\\ &\mu^k = \|g^k\|^2/\|g^{k-1}\|^2. \end{aligned}$$

Here g is the gradient vector. The ith element of the gradient vector is

(5.6) $$g_i = \partial f/\partial \alpha_i.$$

For the modified Newton method, the direction of search is given by

(5.7) $$p^k = -B^{-1}g^k \quad \text{for all } k.$$

Here B is a matrix. The element $b_{i,j}$ of B is given by

(5.8) $$b_{i,j} = \partial^2 f/\partial \alpha_i \partial \alpha_j,$$

under the condition that the second partial derivatives of β and γ with respect to α_i and α_j are assumed to be zero in deriving (5.8). If the second partials of β and γ are not set to zero, the method is the usual Newton method. The result of setting the second partials equal to zero in obtaining B is that the B matrix is always positive definite. This implies that the choice of direction of search given by (5.7) guarantees a descent direction because the directional derivative is negative.

$$(g^k)^T p^k = -(g^k)^T B^{-1} g^k < 0. \tag{5.9}$$

In the above, the minimization is carried out only with respect to the first $n - 1$ elements of the α vector because α_n is kept constant to fix the angular orientation. Therefore, the vectors p and g are of dimension $n - 1$ and the B matrix is of dimension $(n - 1) \times (n - 1)$.

An explanation of the elements of the gradient vector and the B matrix would complete this section. Because we do not want to minimize with respect to α_n, we consider its influence as a constant influence on f and separate it fom the influence of the other elements of the α-vector. Thus, let A' be the matrix A without its last column (hence, A' is of dimension $(n - 1) \times (n - 1)$). Let a be the vector equal to the last column of A. Let δ' be a vector of dimension $n - 1$ consisting of the first $n - 1$ elements of δ. Let γ' be a vector of dimension $n - 1$ consisting of the first $n - 1$ elements of γ. Then

$$\begin{aligned} f = {} & \tfrac{1}{2}(A'\gamma' + \gamma_n a + U_0 - \beta)^T(A'\gamma' + \gamma_n a + U_0 - \beta) \\ & + \tfrac{1}{2}K_v(\delta'^T\gamma' + \gamma_n\delta_n - V_0)^2. \end{aligned} \tag{5.10}$$

Therefore,

$$\frac{\partial f}{\partial \alpha_j} = (A\gamma + U_0 - \beta)^T \left(A' \frac{\partial \gamma'}{\partial \alpha_j} - \frac{\partial \beta}{\partial \alpha_j}\right) + K_v(\delta^T\gamma - V_0)\delta'^T \frac{\partial \gamma'}{\partial \alpha_j}, \tag{5.11}$$

$$\frac{\partial^2 f}{\partial \alpha_i \partial \alpha_j} = \left(A' \frac{\partial \gamma'}{\partial \alpha_i} - \frac{\partial \beta}{\partial \alpha_i}\right)^T \left(A' \frac{\partial \gamma'}{\partial \alpha_j} - \frac{\partial \beta}{\partial \alpha_j}\right) + K_v \delta'^T \frac{\partial \gamma'}{\partial \alpha_i} \delta'^T \frac{\partial \gamma'}{\partial \alpha_j}. \tag{5.12}$$

Programming the gradient vector g and the matrix B is greatly simplified because

$$\frac{\partial \gamma_i'}{\partial \alpha_j} = 0 \quad \text{and} \quad \frac{\partial \beta_i}{\partial \alpha_j} = 0 \quad \text{for } i \neq j.$$

Thus, for example, the derivative of γ' with respect to α_j is a vector with the jth element being the only nonzero element. This implies that

$$A' \frac{\partial \gamma'}{\partial \alpha_j} = \frac{\partial \gamma_j'}{\partial \alpha_j} \quad (j\text{th column of the } A' \text{ matrix}).$$

When these details are taken into account in the programming, the execution time of the computer code is greatly reduced.

Another time saving feature can be used in computing the B matrix. Let us write

$$\text{Term} = \left(A' \frac{\partial \gamma'}{\partial \alpha_i}\right)^T \left(A' \frac{\partial \gamma'}{\partial \alpha_j}\right) = \left(\frac{\partial \gamma'}{\partial \alpha_i}\right) A'^T A' \frac{\partial \gamma'}{\partial \alpha_j}.$$

If we compute and store the transpose product of the A' matrix along with the fundamental solutions, the time saving is obvious. Term is simply evaluated as

$$\text{Term} = \frac{\partial \gamma_i'}{\partial \alpha_i} \frac{\partial \gamma_j'}{\partial \alpha_j} \quad (\text{element } i, j \text{ of stored } A'^T A' \text{ matrix}).$$

6. **Examples.** Examples are presented in this section to illustrate various features of the numerical conformal mapping procedure. Before constructing the specific mappings that serve as examples, the fundamental solutions and the transpose product of A' were computed and stored for three different meshes (different number of mesh points). All the examples made use of these stored quantities. For every example considered, the boundary of the physical domain was described in terms of the cubic splines that define β and γ as functions of α, the length along the curve. The (X, Y) pairs needed to evaluate the spline coefficients were input at discrete points on the boundary, and the arc length was approximated by the chord. The next step was to minimize the residue. Then, the discrete C-R equations were solved with the γ values associated with the optimum α values and the prescribed U_0 to determine the discrete conformal mapping. To display the map the U values at the V mesh points were determined by interpolation. The grid lines in the physical domain corresponding to the constant coordinate lines passing through the V mesh points in the computational domain were plotted for each of the examples. The constant coordinate lines in the computational domain are shown in Figure 3 for a 32×16 mesh ($n = 32, m = 16$).

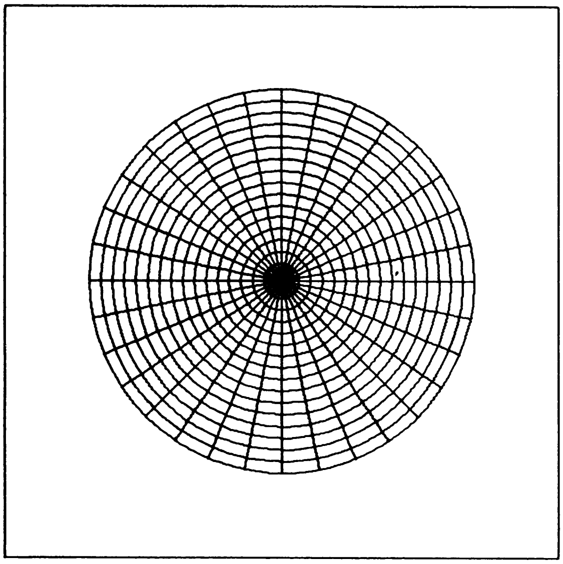

FIGURE 3
Constant coordinate lines in computational domain

To start the minimization procedure for all the examples, α values were chosen at equal increments along the bounding curve unless otherwise stated. The value of α_n was left unchanged throughout the minimization. Several termination criteria were employed. Each subproblem was terminated when the norm of the gradient vector became as small as required. The overall procedure was terminated when the residue became as small as required or when the constraint on V_0 was satisfied to the required degree.

Before discussing the specific examples, some general timing information is pertinent. The times reported are for the Itel AS/5 computer or the IBM 370/158 computer. All operations were performed in double precision arithmetic, and the programs were written in FORTRAN and compiled in H level with OPT = 2. The computing times for the fundamental solutions, the times to compute the direction of search using (5.7) and approximate estimates of the times to perform linear searches are all given in Table 2 for the three meshes.

TABLE 2
General timing information

n	m	Total time for n fundamental solutions (seconds)	Average time for 1 fundamental solution (seconds)
16	16	3.15	0.197
32	32	23.12	0.722
64	32	103.94	1.624

n	Time to solve (5.7) (seconds)	Time for linear search (seconds)
16	0.01	0.05
32	0.07	0.17
64	0.48	0.45

Both the CG1 and the modified Newton methods use linear searches. In addition, the latter method has to solve the system of equations in (5.7). This makes the Newton method slower per iteration than the CG1 method. But in many cases the Newton method takes fewer steps to converge, and the overall time is smaller. To solve (5.7) a linear equation solver that exploits the symmetry of the B matrix has been used and is recommended. The modified Newton method is especially effective near the absolute minimum. If the B matrix is singular or possesses a very small determinant, the Newton method cannot be used. The CG1 method does not have any such restrictions.

The CG1 method has to be restarted with the steepest descent step after n iterations at most. The Newton method needs no restarting.

The first example illustrates that changing the point that is mapped into the center of the unit disk changes the conformal mapping for the same physical domain. In addition to showing the constant coordinate lines in the computational domain, Figure 3 also portrays the constant coordinate lines in physical space for a unit disk mapped onto the unit disk with the center of the physical domain mapped into the center of the unit disk. Figure 4 shows the constant coordinate lines in the physical domain when U_0 was chosen to be 0.2 rather than zero. V_0 was left unconstrained for this example. Thus there was only one subproblem to solve, and the penalty constant was set to zero. For this example, V_0 turned out to be 0.335 at the end of minimization. Six iterations and 2 seconds sufficed with the Newton method for $n = 32$. The residue was $O(10^{-9})$ at termination.

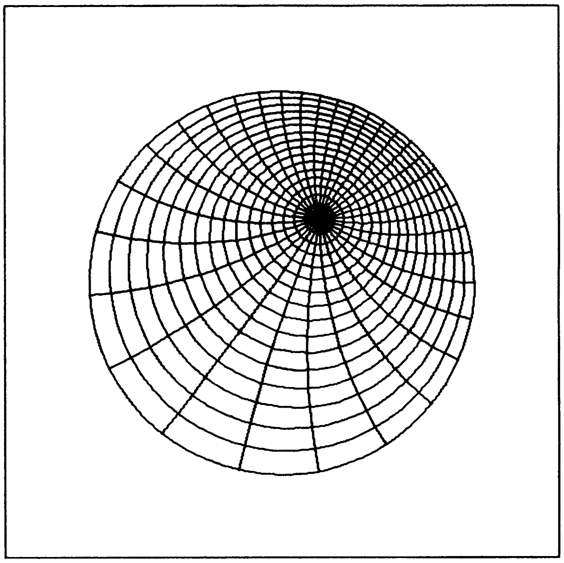

FIGURE 4

Conformal mapping of unit disk with $U_0 = 0.2$, $V_0 = 0.335$

The second example illustrates the use of the penalty function technique to constrain V_0. Again, the unit disk was mapped onto the unit disk. But now the point given by $U_0 = 0.0$ and $V_0 = 0.2$ was chosen to map into the center. The penalty constant was defined to be 4.0 for the first subproblem and was increased in multiples of 4.0 between subproblems. Three subproblems sufficed for convergence with the total number of iterations being 18 for the Newton method and the time for computation being 5 seconds ($n = 32$). The residue was $O(10^{-6})$ and the constraint on V_0 was satisfied to the same order. Figure 5 displays this example.

The third example is the conformal mapping of an ellipse with a semimajor axis of 1.25 and a semiminor axis of 0.75. The regions of high curvature at the extremes of the X axis required a mesh with $n = 64$ for accuracy. Both U_0 and V_0 were set to

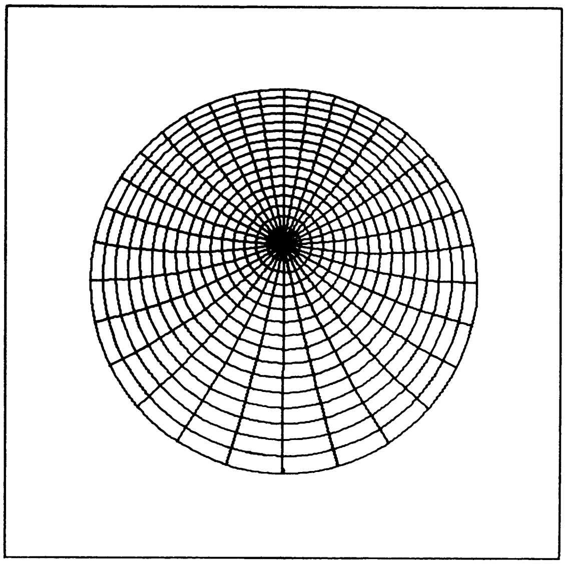

FIGURE 5
Conformal mapping of unit disk with $U_0 = 0.0$, $V_0 = 0.2$

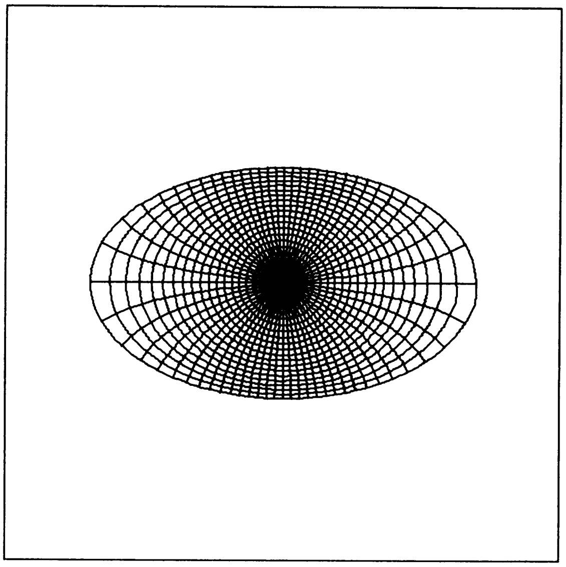

FIGURE 6
Conformal mapping of ellipse

zero for this case. A mesh with $n = 32$ was tried, and the method converged in 6 seconds after 3 subproblems and 21 iterations to $R = O(10^{-3})$. The solution for a mesh with $n = 64$ converged to $R = O(10^{-5})$ in 18 seconds after 3 subproblems and 15 iterations when the iterations were started with the usual starting α values. But when the iterations for $n = 64$ were started from α values obtained by interpolation from the $n = 32$ solution, the method converged in 7 iterations and 8 seconds. This example illustrates the time saving that occurs when lower order solutions are used to start the iterations for a mesh with more points (for this example the time saved is 4 seconds). The modified Newton method was used for this example, and the constant coordinate lines are shown in the physical domain in Figure 6.

The fourth example is the conformal mapping of a rock (an arbitrarily drawn simply connected domain without corners) onto the unit circle. This illustrates the generality of the method. For this example V_0 was left unconstrained. Convergence to a value of the residue of $O(10^{-7})$ was reached in only 6 iterations and 2 seconds with the modified Newton method ($n = 32$). Figure 7 displays this example.

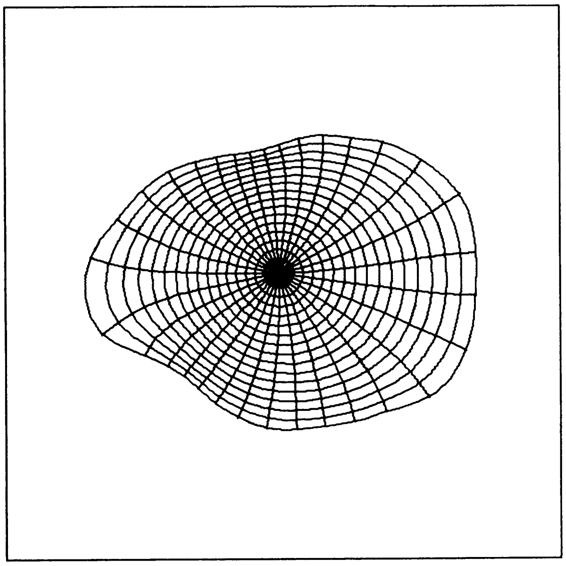

FIGURE 7
Conformal mapping of rock

The last two examples illustrate the use of the modified Karman-Trefftz transformation to trèat boundaries with corners. The Karman-Trefftz transformation is given by

$$\frac{\tau - 1}{\tau + 1} = \left(\frac{z - h}{z + h^*}\right)^{\delta}.$$

·The complex z plane is the orgiinal physical plane where the bounding curve has a corner at h. The complex τ plane is the intermediate physical domain, where the corner has been smoothed out by the proper choice of δ. The example chosen is the region bounded by the lemniscate of Bernoulli shown as the bounding curve in Figure 8. The included angle at the corner is a right angle, and δ was set to 2 to remove the corner.

The forward Karman-Trefftz transformation removed the corner and mapped the lemniscate onto a near circle shown as the bounding curve in Figure 9. The conformal mapping from the near circle onto the unit disk was performed numerically, and the constant coordinate lines are displayed in Figure 9. Four iterations with CG1 method

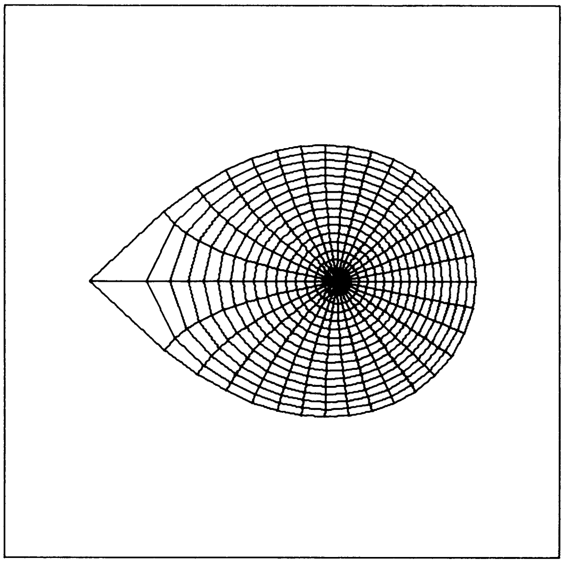

FIGURE 8
Conformal mapping of lemniscate of Bernoulli using Karman-Trefftz transformation

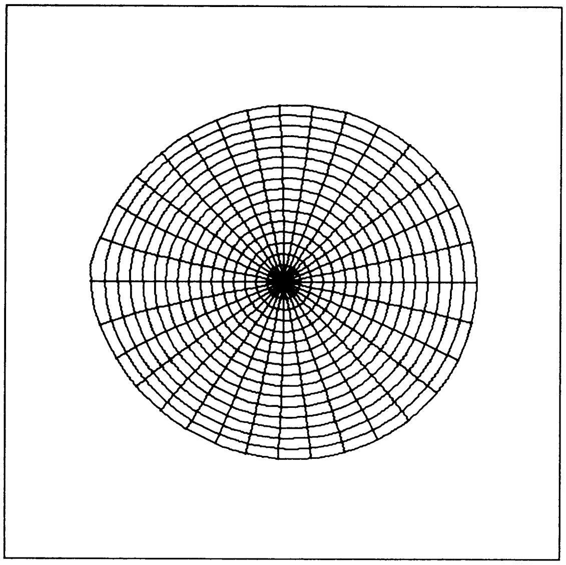

FIGURE 9
Conformal mapping of near circular intermediate domain

sufficed when V_0 was unconstrained, and the time required was just one second for a mesh with $n = 32$. The modified Newton method failed for this example because the determinant of the B matrix was very small. The points of intersection were mapped back onto the original z plane, and Figure 8 shows the constant coordinate lines for the conformal mapping of the lemniscate of Bernoulli.

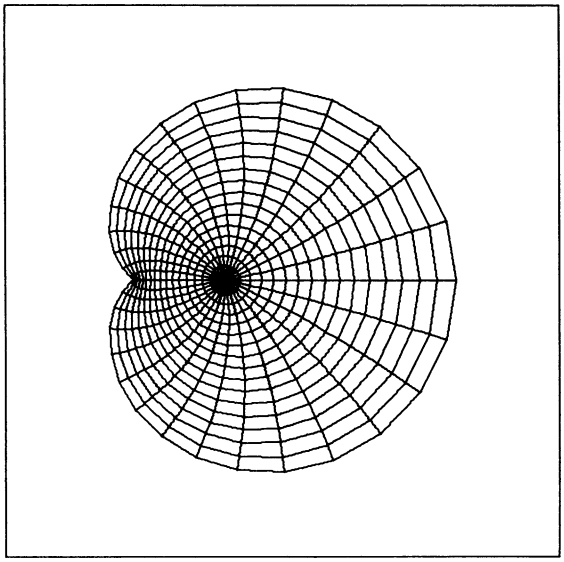

FIGURE 10

Another conformal mapping using Karman-Trefftz transformation

Another example using the Karman-Trefftz transformation was obtained by performing another forward transformation from the mapping shown in Figure 9 and the resulting constant coordinate lines are shown in Figure 10. We could have started from the bounding curve of Figure 10, done a Karman-Trefftz transformation to obtain a near circle, performed the numerical conformal mapping of the near circle and mapped back to Figure 10. For these last two examples, the constant coordinate lines in the physical domain have been plotted with straight line segments to preserve the accuracy of the physical domain at the corner, but this might be misleading in the interior.

7. **Concluding Remarks.** A new, general method has been presented to compute the discrete conformal mapping of any simply connected region onto the unit disk. The direct relationship between the Riemann Mapping Theorem and the method has been brought out in the presentation. The speed of the method has been demonstrated by the examples. The residue R is an estimate of the global error of the numerical method and is obtained as part of the computation. Theodorsen's and Barfield's methods do not possess such a convenient error estimate. Moretti's technique is entirely analytic and so does not possess any discretization error. It is very well suited for generating computational grids for complex geometries, but does not address the general problem of finding the conformal mapping of a given domain onto the unit disk. Theodorsen's method is mainly suited for airfoil-like geometries, and Barfield's method is slow.

Barfield's direct method requires the solution of a linear system of equations. Every different physical domain gives rise to a new system of equations. Thus, Barfield's method would not be suitable for problems where the shape of the physical domain changes in time. On the other hand, the fundamental solutions of the present iterative method have to be constructed only once. The iterations for the solution at a new time could use the previous time level solution as a starting point, thus making the present method an efficient one for time varying physical domains. In its present form, the new method maps the interior of the region onto the interior of the unit disk. It is an easy extension with the help of preliminary analytic conformal transformation formulas to map the exterior of a domain onto the interior of the disk. Extension of the method to multiply connected regions needs more study.

A possible disadvantage of the new method is that the minimization techniques cannot guarantee that the absolute minimum $R = 0$ will be attained. However, they do guarantee that a relative minimum corresponding to a small value of the residue will be reached. This has proved to be adequate for the varied examples considered. The matrix multiplications needed to compute the residue are another disadvantage for meshes with a very large number of mesh points due to possible error accumulation. It was necessary to compute the U values at the boundary points when the V values were prescribed. This was done in terms of a matrix multiplication where the coefficient matrix comprised the fundamental solutions. This could also be done very rapidly by using matrix decomposition Poisson solvers [3] to solve for the U and V values at the few points near the boundary used in the interpolation formulas. Matrix decomposition Poisson solvers are especially sutiable when it is necessary to compute the solution to the discrete Poisson equations at only selected grid rows or columns. If this approach is feasible, it would altogether eliminate matrix multiplications from the numerical conformal mapping procedure.

Department of Aerospace Engineering and
Engineering Research Institute
Iowa State University
Ames, Iowa 50011

1. I. H. ABBOTT & A. E. VON DOENHOFF, *Theory of Wing Sections*, Dover, New York, 1958.

2. W. D. BARFIELD, "Numerical method for generating orthogonal curvilinear meshes," *J. Computational Phys.*, v. 5, 1970, pp. 23–33.

3. B. L. BUZBEE, "A capacitance matrix technique," in *Sparse Matrix Computations* (Proc. Sympos., Argonne National Laboratory, 1975), Academic Press, New York, 1976, pp. 365–373.

4. J. B. CONWAY, *Functions of One Complex Variable*, Graduate Texts in Mathematics, no. 11, Springer-Verlag, New York, 1975.

5. A. V. FIACCO & G. P. McCORMICK, *Nonlinear Programming: Sequential Minimizing Techniques*, Wiley, New York, 1968.

6. R. FLETCHER & C. M. REEVES, "Function minimization by conjugate gradients," *Comput. J.*, v. 7, 1964, pp. 149–154.

7. H. LOMAX & E. D. MARTIN, "Fast direct numerical solution of the nonhomogeneous Cauchy-Riemann equations," *J. Computational Phys.*, v. 15, 1974, pp. 55–80.

8. G. MORETTI, "Conformal mappings for computations of steady, three-dimensional, supersonic flows," in *Numerical/Laboratory Computer Methods in Fluid Mechanics* (Papers presented at the winter meeting of the ASME, New York, 1976), ASME, New York, 1976.

9. M. J. D. POWELL, "A survey of numerical methods for unconstrained optimization," *SIAM Rev.*, v. 12, 1970, pp. 79–97.

10. R. A. SWEET, "A generalized cyclic reduction algorithm," *SIAM J. Numer. Anal.*, v. 11, 1974, pp. 506–520.