JOURNAL OF COMPUTATIONAL PHYSICS 8, 392–408 (1971)

Development of a General Finite Difference Approximation for a General Domain
Part I: Machine Transformation

WEN-HWA CHU

Department of Mechanical Sciences, Southwest Research Institute, San Antonio, Texas 78228

Received January 7, 1971

The general idea of a "numerical transform" on a high speed computer for a general domain, abbreviated as a machine transformation, is illustrated by employing an equilateral triangle mesh plane for a general, second-order quasi-linear elliptic partial differential equation subject to a general third boundary value condition in a general domain. The feasibility of the technique for linear elliptic equations is demonstrated by two test problems, for which the numerical solutions are compared with exact analytic solutions. A new computing technique is devised for linear elliptic equations, and possible extensions to quasi-linear boundary value problems are discussed.

This method eliminates the programming difficulties of a finite-difference method near the curved boundaries, and, most important, it can be preprogrammed for a general class of domains to yield numerical solutions.

I. INTRODUCTION

The method of conformal transformation is one of the most useful tools in classical analysis [1]. However, the method is not practical analytically for non-Laplacian operators or three-dimensional domains because of the difficulties in solving the differential equations after the transformation. Furthermore, some transformations are difficult to find and are not included in a dictionary [2].

In this paper, we shall consider a "numerical transformation," using high speed computers, which is abbreviated as a "machine transformation" using the terminology of Ref. [3]. It can be shown with ease that a second-order equation does not change type under a general transformation, nor does a system of second-order differential equations.

The basic idea is a generalization of the so-called "logical plane" used by Winslow [4] as the transform plane.[1] The logical plane was employed by Winslow

[1] This is a more useful idea than the ideas given in Ref. [5], which are limited to "conformal mappings," which map a given domain into a rectangle of a fixed aspect ratio. They map only discrete points to discrete points.

to generate a "topologically regular" nonuniform triangular mesh in the physical plane. His finite difference scheme as derived is limited to quasi-linear Poisson equations since, as in Ref. [6], the divergence theorem is applied. However, his approach may reduce the computing time in cases where there are singular corners.

Winslow's logical plane is composed of equilateral triangle meshes for which simple finite difference equations can be preprogrammed with relative ease, and it can be applied to a whole class of domains. It requires as input only (discrete) specified boundary points to generate the finite-difference equations. This flexibility is often sought in practical computer programs. Winslow first applied his method to some magnetostatic problems. It was later extended successfully by Chu to compute the low-gravity liquid sloshing in arbitrary axisymmetric rigid tanks [7].[2] A single parallelogram logical plane could be used for an arbitrary convex interior domain, which contained no confluent angles and no more than four sharp corners. (Domains of equilateral triangles of more corners could be generated if necessary, and re-entrant corners might be mapped onto straight edges with a corner formulation.)

The basic principle of the transformation is well known and its practicality will be illustrated by two-dimensional test examples. To demonstrate the generality of the methods, an "oblique transformation" of a rectangular or an elliptic domain into an equilateral triangle mesh plane will be made (Fig. 1); the transformation is, in general, not conformal [cf. Eqs. (9a, b)]. This plane (Winslow's "logical plane") is one of the many possible transform planes which can be used.

In short, the purpose of the paper is to present the basic idea of a "machine transformation" for partial differential equations, particularly, of the elliptic type, in a general bounded domain and to demonstrate its versatility and practicality on a high speed computer.

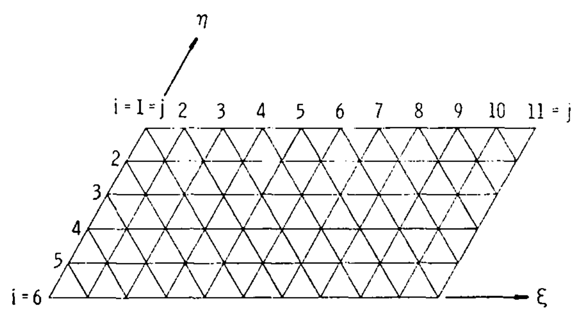

FIG. 1. Transform plane with a 6 × 11 equilateral triangle mesh ($M = 6$, $N = 11$).

[2] See also Concus, P., Crane, G. E. and Satterlee, H. M., "Small Amplitude Lateral Sloshing in Spheroidal Containers Under Low Gravity Conditions," NASA CR 72500, February 1969.

II. Mathematical Transformation

Two-dimensional elliptic boundary value problems are considered. The general transformation from the physical plane (x, y) to the transformed plane (ξ, η) is

$$\xi = \xi(x, y), \qquad \eta = \eta(x, y), \tag{1a, b}$$

and the inverse transform is

$$x = x(\xi, \eta), \qquad y = y(\xi, \eta). \tag{2a, b}$$

The Jacobian of the transformation [8] is

$$J = J\left(\frac{x, y}{\xi, \eta}\right) = x_\xi y_\eta - x_\eta y_\xi \neq 0, \tag{3}$$

where

$$x_\xi = \frac{\partial x}{\partial \xi}, \qquad y_\eta = \frac{\partial y}{\partial \eta}, \quad \text{etc.} \tag{4a, b}$$

It is easy to show that

$$\xi_x = \frac{y_\eta}{J}, \qquad \xi_y = \frac{-x_\eta}{J}, \qquad \eta_x = \frac{-y_\xi}{J}, \qquad \eta_y = \frac{x_\xi}{J}. \tag{5a, b, c, d}$$

Then

$$\frac{\partial}{\partial x} = \left(\frac{y_\eta}{J}\frac{\partial}{\partial \xi} - \frac{y_\xi}{J}\frac{\partial}{\partial \eta}\right), \tag{6a}$$

$$\frac{\partial}{\partial y} = \left(\frac{x_\eta}{J}\frac{\partial}{\partial \xi} + \frac{x_\xi}{J}\frac{\partial}{\partial \eta}\right). \tag{6b}$$

Higher derivatives can be obtained by repeated operations.

Two tasks of transformation are involved: one is to find the interior physical points after specifying the physical boundary at a number of discrete points, and the other is to transform the partial differential equations into the new variables before being approximated by the finite-difference equations.

A. *Mapping of the Physical Domain*

The choice of this mapping is largely dependent on its simplicity and the effort required for a desired accuracy.

Without loss of generality, a simply-connected domain with no more than four sharp corners will be considered [3]. The boundary of the physical domain is specified at discrete points (x_b, y_b). These points correspond to the boundary points (ξ_b, η_b) in the transform plane (ξ, η). Since we desire to have a prescribed, convenient mesh in the (ξ, η) plane, (ξ, η) must be used as independent variables; their values are governed by any suitable elliptic partial differential equation as a first-boundary value problem. The simplest choice appears to be the "equipotentials" transformation [4]; i.e., ξ, η must satisfy the Laplace equation in the physical plane:

$$\nabla^2\xi = 0, \qquad \nabla^2\eta = 0, \tag{7a, b}$$

where

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \tag{8}$$

The dependent and independent variables can be interchanged by applying Eqs. (5a, b, c, d) and (6a, b). One finds that

$$a x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} = 0 \tag{9a}$$

$$a y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} = 0 \tag{9b}$$

where

$$a = x_\xi^2 + y_\xi^2, \qquad \beta = x_\xi x_\eta + y_\xi y_\eta, \qquad \gamma = x_\eta^2 + y_\eta^2 \tag{10}$$

Equations (9a, b) are clearly coupled quasi-linear elliptic equations, and only in special cases ($x_\xi = y_\eta$ and $x_\eta = -y_\xi$) can they be reduced to Laplace equations, for which mapping is conformal.

Equations (9a, b) in the general case can be conveniently solved by the finite difference method [4, 9, 10] with successive overrelaxation[3] (SOR) of the dependent variables and underrelaxation of the coefficients with linearly interpolated initial guess. An intermediate solution can be constructed by SOR, so that each center value is the mean value of the six neighboring points. This will then be a better initial guess [4] to the true solution. The discrete values of (x, y) at the corresponding net point (ξ, η) are thus determined. The "equivalent finite-difference net" in the transformed plane is used simply as auxiliary lines for the finite-difference technique, but also it can be used to see if the "machine transformation" yields a reasonable set of finite difference equations in the physical plane.

In general, the numerically calculated boundary may not coincide completely with the given boundary at intermediate points. This is not an essential limitation since only discrete points are needed in the basic principle of the finite-difference method. The finer the mesh, the smaller would be the numerical error.

[3] Cf. Appendix B.

B. *Mapping of the Differential Equation*

For this paper, a general, second-order quasi-linear elliptic partial differential equation will be transformed. In the physical plane, the equation takes the form

$$A \frac{\partial^2 \phi}{\partial x^2} + B \frac{\partial^2 \phi}{\partial x\, \partial y} + C \frac{\partial^2 \phi}{\partial y^2} + D \frac{\partial \phi}{\partial x} + E \frac{\partial \phi}{\partial y} + F = \mathscr{L}\phi + F = 0, \quad (11)$$

where A, B, C, D, E, F are functions of x, y, and ϕ, and

$$AC > B^2/4 \text{ (elliptic).} \quad (11a)$$

A generalized, third-boundary value problem is considered:

$$\alpha_1 \frac{\partial \phi}{\partial x} + \alpha_2 \frac{\partial \phi}{\partial y} + \alpha_3 \phi = \alpha_4 , \quad (12)$$

which contains the first, second, and third boundary-value problems [11]. The coefficients α_1, α_2, α_3, α_4 may be functions of x, y.

Extensions to simultaneous, second-order quasi-linear partial differential equations can be readily made. Generalizations of the boundary conditions may also be possible.

Using Eqs. (6a, b), Eqs. (11) and (12) can be transformed easily to the form[4]

$$\bar{A} \frac{\partial^2 \phi}{\partial \xi^2} + \bar{B} \frac{\partial^2 \phi}{\partial \xi\, \partial \eta} + \bar{C} \frac{\partial^2 \phi}{\partial \eta^2} + \bar{D} \frac{\partial \phi}{\partial \xi} + \bar{E} \frac{\partial \phi}{\partial \eta} + \bar{F} = 0 \text{ in } \bar{S} \quad (13)$$

and

$$\bar{\alpha}_{1j} \frac{\partial \phi}{\partial \xi} + \bar{\alpha}_{2j} \frac{\partial \phi}{\partial \eta} + \bar{\alpha}_{3j} \phi = \bar{\alpha}_{4j} \text{ on } \partial \bar{S}, \quad (14)$$

where $j = 1, 2, 3, 4$ corresponds to top, right, lower, and left boundaries (Fig. 1), respectively. Analogous extension to simultaneous differential equations can also be made.

For completeness, $\bar{A}$, $\bar{B}$, $\bar{C}$, $\bar{D}$, $\bar{E}$, $\bar{F}$ are given in Appendix A.

III. Method of Solution

The general principle is demonstrated by employing the equilateral triangle mesh plane as discussed in the Introduction and by using the finite-difference technique.

[4] In general, $4\bar{A}\bar{C} - \bar{B} = 4AC - B$, i.e., the differential equation does not change type.

A. *Difference Equation Using Equilateral Triangle Mesh*

1. *Interior Nodes.* For an interior node-0, the first and second derivatives of a function f (which can be ϕ, x, y, etc.) can be expressed in terms of six neighboring points and the center point (Fig. 2) with a second-order truncation error as

$$f_{\xi 0} = \tfrac{1}{6}[(f_1 + 2f_6 + f_5) - (f_2 + 2f_3 + f_4)], \quad (15a)$$
$$f_{\eta 0} = \tfrac{1}{6}[(f_2 + 2f_1 + f_6) - (f_3 + 2f_4 + f_5)], \quad (15b)$$
$$f_{\xi\xi 0} = f_6 - 2f_0 + f_3\,, \quad (15c)$$
$$f_{\eta\eta 0} = f_1 - 2f_0 + f_4\,, \quad (15d)$$
$$f_{\xi\eta 0} = \tfrac{1}{2}[(f_1 + f_6 + f_3 + f_4) - (f_2 + f_5 + 2f_0)]. \quad (15e)$$

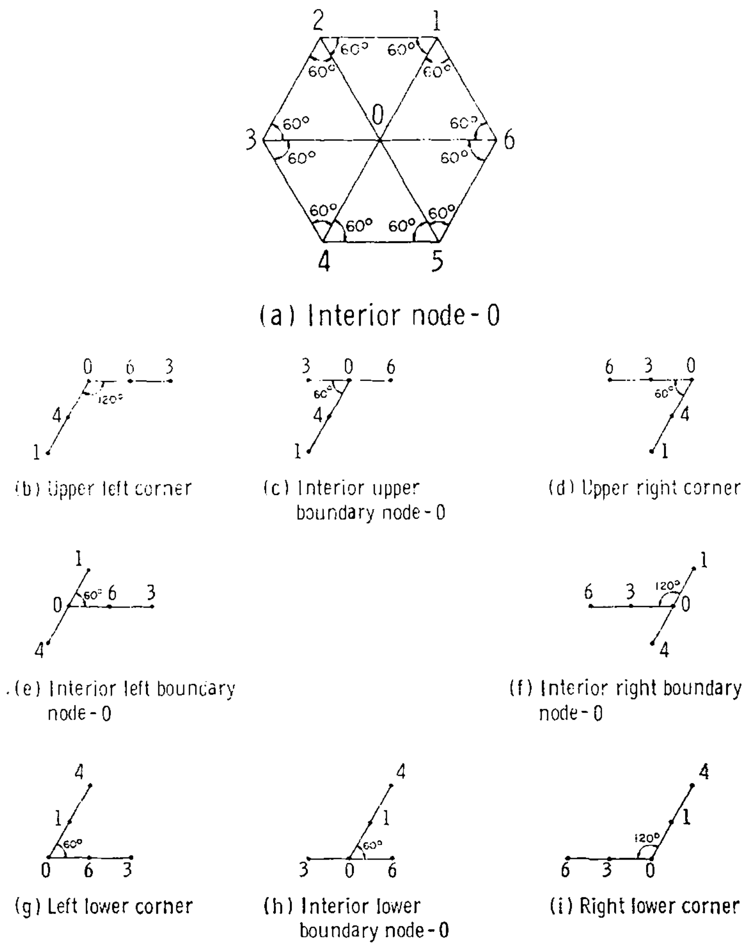

FIG. 2. Notations for the neighboring points around node-0 in transform plane.

The approximation for the first derivatives is not unique, since the determinant vanishes for the Taylor expansions of six neighboring points, each of which is truncated to the second order. However, using six neighboring points as above [4] may give more accurate approximations on the average. The governing difference equation for an interior point is omitted here for brevity but is included in Appendix B for completeness.

2. *Interior Boundary Nodes.* For the interior boundary nodes, central difference formulas can be used for tangential derivatives and forward or backward formulas for the "inward" derivatives; e.g.,

(1) For an interior left boundary node (Fig. 2e),

$$\left(\frac{\partial f}{\partial \xi}\right)_0 = -\frac{1}{2}(3f_0 - 4f_6 + f_3), \tag{16a}$$

$$\left(\frac{\partial f}{\partial \eta}\right)_0 = \frac{1}{2}(f_1 - f_4). \tag{16b}$$

(2) For an interior upper boundary node (Fig. 2c),

$$\left(\frac{\partial f}{\partial \xi}\right)_0 = \frac{1}{2}(f_6 - f_1), \tag{17a}$$

$$\left(\frac{\partial f}{\partial \eta}\right)_0 = \frac{1}{2}(3f_0 - 4f_4 + f_1). \tag{17b}$$

At other interior boundary nodes, the expressions are analogous. The above equations are all of the second-order truncation error. Forward and backward formulas are known to possess larger truncation errors.

3. *Corner Nodes.* For corner nodes, forward and backward formulas on both sides need to be used[5]; hence, larger truncation errors result unless finer space meshes are prescribed near the corners. The explicit equations for the first derivatives at the upper left corner are simply Eqs. (16a) and (17b).

For the upper left corner (Fig. 2b), say, the average of Eq. (14) with $j = 1$ and 4 is used. It is possible that, for a nonzero flux, a weighted average might be required. Thus,

$$(\bar{\alpha}_{11} + \bar{\alpha}_{14})\left(\frac{\partial \phi}{\partial \xi}\right)_0 + (\bar{\alpha}_{21} + \bar{\alpha}_{24})\left(\frac{\partial \phi}{\partial \eta}\right)_0 + (\bar{\alpha}_{31} + \bar{\alpha}_{34})\,\phi_0 = (\bar{\alpha}_{\ 1} + \bar{\alpha}_{44}). \tag{18}$$

Similar equations hold for the other corners.

[5] Techniques such as extrapolating to the exterior and then using central difference formulas for the boundary condition in the governing interior point difference equation may not be applicable due to the need of diagonal exterior nodes.

It is noted that, if the boundary condition is $\partial\phi/\partial n = 0$, the consistent normals must be used, otherwise the diagonal matrix coefficient would vanish and the usual SOR method cannot be applied.

B. *Jacobian of Transformation*

The finite-difference approximations for the derivatives used are second-order approximations in the sense that the truncation error is of the second order of the maximum local mesh size. The resultant finite difference equation is a second-order approximation if the Jacobian of transformation J possesses an error of the second order. To check the retention of second-order accuracy, the maximum percentage change from the maximum of $x_\xi y_\eta$ and $x_\eta y_\xi$ to $x_\xi y_\eta - x_\eta y_\xi$ may be spot checked, if we know the values of x, y at the nodes. In our test cases, analytic solutions were used to check the accuracy of the solution, and the values of J were printed over the mesh. Of particular interest is the Jacobian at the corners of the transformed plane. Although the accuracy of the corner point Jacobian will not affect either a first boundary-value problem or a second boundary-value problem, it *may* affect a general third boundary-value problem. If the physical boundary is smooth, the corner point Jacobian would become small and the chance of losing accuracy becomes large. For example, J lost one significant figure in the case of a full ellipse with $M = 11$, $N = 21$. In general, this may mean a reduction from a second-order process to a first-order process. However, when the Jacobian at a corner is "very small" due to the corresponding physical corner being smooth (180 degrees), the numerator above the Jacobian or its highest power dominates, and then the loss of significant figures in the Jacobian at the corner may not reduce the order of accuracy. Except when the mapping is conformal, a general corner point analysis seems to be very difficult [cf. Eqs. (9a, b)]. If singular corners are suspected and approximate numerical solutions can be obtained in the physical plane without using the corners, then modified finite difference forms without using corner points in the neighborhood of the corners can be programmed to achieve the same goal. With the same total number of nodes, finer meshes near the physical corners (which might be fictitious if the boundary is a smooth curve) would possibly reduce the maximum errors when occurring there.

C. *Higher Order Approximations and Other Transformations*

Higher-order finite difference formulas may be generated with an increase in programming effort. Such formulas might not be very effective in reducing truncation errors in the boundary.

In some cases, a different transformed domain $\bar{S}$ may be required to avoid large distortion and to have an accurate Jacobian.

Unless the very time-consuming large matrix inversion process is required as part of a larger program, it is preferable to refine the mesh size somewhat analogous to the popular Simpson's rule versus Weddle's rule or the Gaussian quadrature formula, etc. [12].

It is emphasized that other transformations have not yet been applied but could be selected in principle. In particular, a square mesh transform should be tried for "reflectively" symmetric problems, since finite difference equations for an equilateral triangle mesh transformation is not reflectively symmetric and yields a slightly nonsymmetric solution for symmetric problems.

D. *Successive Overrelaxation* (SOR) *Procedure* [13]

After mapping to the transformation plane, the corresponding discrete values of (x, y) at the nodes can be solved by overrelaxation of the dependent variables (Appendix B) and underrelaxation of the quasi-linear coefficients. This is called the "mesh generation," although a mesh in the physical plane is not essential except to locate the nodes where the values of the solution are computed.

For elliptic differential equations, an overrelaxation factor between 1 and 2 is usually used; this is based on the convergence analysis of restricted classes of equations. For p-cyclic matrices, the optimum relaxation factor was given in Reference [14]. In particular, if two-cyclic, then

$$\omega_{\text{opt}} = \frac{2}{1 + (1 - \mu^2)^{1/2}} . \tag{19}$$

If $\omega < \omega_{\text{opt}}$, then the eigenvalue of the associated block Jacobi matrix is

$$\mu = \frac{\omega + \lambda - 1}{\omega \sqrt{\lambda}} . \tag{20}$$

The approximate convergence rate or eigenvalue λ is

$$\lambda^{(n)} \cong \left[\frac{\sum_i \sum_j (\phi_{ij}^{(n+1)} - \phi_{ij}^{(n)})^2}{\sum_i \sum_j (\phi_{ij}^{(n)} - \phi_{ij}^{(n-1)})^2}\right], \; n > 1. \tag{21}$$ [6]

Winslow [4] found that, for certain linear as well as nonlinear problems, a satisfactory "optimizing" scheme is

$$\mu = \frac{\omega^{(n)} + \lambda^{(n)} - 1}{\omega^{(n)} \sqrt{\lambda^{(n)}}} , \tag{22a}$$

$$\omega'_{\text{opt}} = \frac{2}{1 + (1 - \mu^2)^{1/2}} - \omega_0 , \tag{22b}$$

$$\omega^{(n+1)} = \beta\omega'_{\text{opt}} + (1 - \beta)\, \omega^{(n)}, \tag{22c}$$

$$\omega_0 \cong 0.01, \qquad \beta \cong 0.05. \tag{22d}$$

[6] For mesh generation, similar expressions are used, treating x and y as uncoupled.

ω_0 permits ω to decrease in nonlinear problems when necessary. If $\lambda = 1$, ω is held constant in our computer programs.[7] Winslow's equation matrix is block tridiagonal [7, p. 243] and is a two-cyclic matrix [13, p. 102] as its associated block Jacobian matrix is weakly two-cyclic. Since diagonal square matrices need not be of the same order, the equation matrix in most cases can also be put in the form of a tridiagonal matrix. It may be a difficult task to find a strictly valid optimum ω for *point relaxation*. Equations (22a, b) for two-cyclic block relaxation were adapted in our SOR procedure for the two test problems, but were abandoned for obtaining the solution given in Ref. [16], as an ω in the neighborhood of unity was needed here for convergence.

Convergence is not guaranteed, especially if the diagonal dominance condition [10, 13] is not satisfied. Nevertheless, revised procedures may lead to a convergence process for practical use. For example, a direct-and-reverse relaxation procedure for linear second-order elliptic equations with selected underrelaxation terms [17] may be used to insure convergence.

A convergence measure E_M for mesh generation and an error measure E_S for solution generation were printed out for guidance;

$$E_M = \left[\frac{\sum_i \sum_j f_{ij}^{(n+1)} - f_{ij}^{(n)^2}}{\sum_i \sum_j f_{ij}^{(n+1)^2}}\right]^{1/2}_{\substack{f=x \text{ or } y,\\ \text{respectively}}} = \left[\frac{(\omega^{(n)})^2 \sum_i \sum_j [R_{f_{ij}}^{(n)}]^2}{\sum_i \sum_j f_{ij}^{(n+1)^2}}\right]^{1/2}, \quad (23), (24)$$

$$E_S = \left[\frac{\sum_i \sum_j R_{\phi_{ij}}^{(n)^2}}{\sum_i \sum_j (\phi_{ij}^{(n+1)})^2}\right], \quad (25)$$

where $R_{g_{ij}}^{(n)}$ is the direct local weighted residual in the n-the iteration of the g-equation, g being x, y, ϕ, respectively. If E_S is defined analogous to E_M for $1 \leqslant \omega < 2$, it will be a factor of one to four larger. E_M and E_S less then 10^{-6} or smaller are usually required, especially when more than one vector component at each node is present (simultaneous partial differential equations).

[7] This procedure is recommended by Winslow. However, it appears to the author that for a general problem a possible advantageous provision may be added so that if $\omega^{(n+1)} < 1$ (which is possible as $\omega_0 > 0$), it should be set to unity. The process reduces to Gauss–Seidel's method of successive displacement [7, p. 236], the convergence of which requires either diagonal dominance and irreducibility of the equation matrix or positive definiteness. After λ reduces again, new $\omega^{(n+1)} > 1$ may be obtained to give a possibly higher rate of convergence. For $(0 < \omega < 2)$, the convergence is only assured if the equation matrix is positive definite [7, p. 261], which means all eigenvalues are positive [15, p. 28].

IV. Test Cases

A. *Torsion Problem for a Beam of Elliptic Cross Section* [18]

The governing equation for the reduced stress function ϕ is

$$\nabla^2 \phi = 0. \tag{26}$$

The boundary condition is

$$\phi = \tfrac{1}{2}(x^2 + y^2) \tag{27}$$

on the boundary

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{28}$$

In the example, $a = 2$, $b = 1$ were used.

The exact solution available for comparison is

$$\phi = \frac{1}{2} \frac{a^2 - b^2}{a^2 + b^2} (x^2 - y^2) + \frac{a^2 b^2}{a^2 + b^2}. \tag{29}$$

First, discrete points on the bounding ellipse are mapped onto the uniformly-spaced bounding parallelogram. This is a first-boundary value problem in the transform plane. The fictitious physical corner points are taken at $x = \pm\frac{3}{4}a$, $y = \pm b\sqrt{\frac{7}{16}}$. On the top and bottom arcs between $-\frac{3}{4} \leqslant x \leqslant \frac{3}{4}$, the x distribution is specified linearly at N points with the corresponding y given by Eq. (28). On the left and right arcs, y is specified linearly at M points with the corresponding x given by Eq. (28).

Numerical solutions were obtained for a 7×11 parallelogram and an 11×21 parallelogram. The maximum local fractional error for the 7×11 mesh was 1.481×10^{-3} or 1.480×10^{-3} at (2, 8) or (6, 4), respectively. For the 11×21 mesh, the errors near the corresponding location were 2.72×10^{-4} at (4, 14) or (8, 7). The asymmetry of the solution was negligible.

With convergence criteria $E_M^- < 10^{-6}$, $E_S < 10^{-6}$, the total CP time was only 5.75 sec for the 7×11 net, and 9.81 sec for the 11×21 net on a CDC-6600 computer. For mesh generation, the number of iterations were 88 and 90 for the initial approximation, and 33 and 36 for the final approximation in the two cases, respectively.

Next, the same problem was solved for a 1/4 upper right domain, with mixed first- and second-boundary conditions. On the top and left boundary of the parallelogram in the transform plane, Eq. (27) holds, while, by symmetry on the left boundary ($j > 1$),

$$\frac{\partial \phi}{\partial x} = 0 \tag{30}$$

and, on the bottom boundary ($i = M$),

$$\frac{\partial \phi}{\partial y} = 0. \tag{31}$$

With a 6×11 parallelogram in the transformed plane with equilateral triangle meshes, the physical plane mesh is shown in Fig. 3.

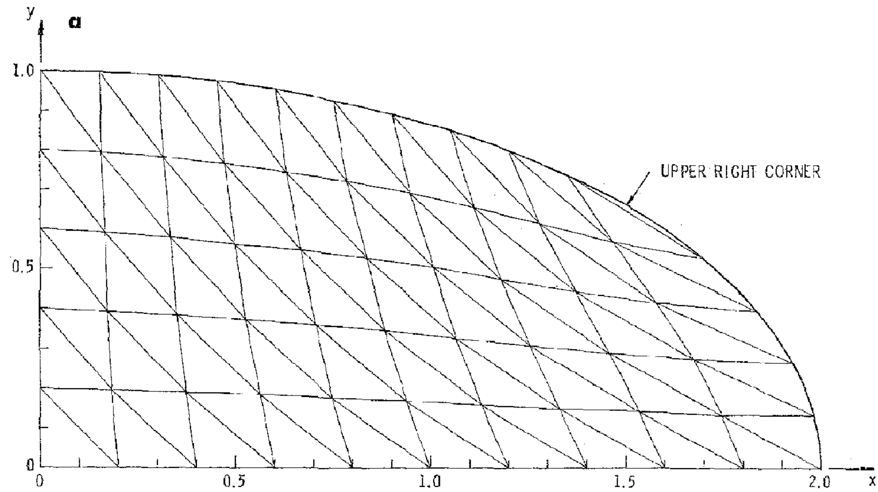

FIG. 3a. Auxiliary mesh in the physical plane of a 1/4-ellipse; $M = 7$, $N = 11$.

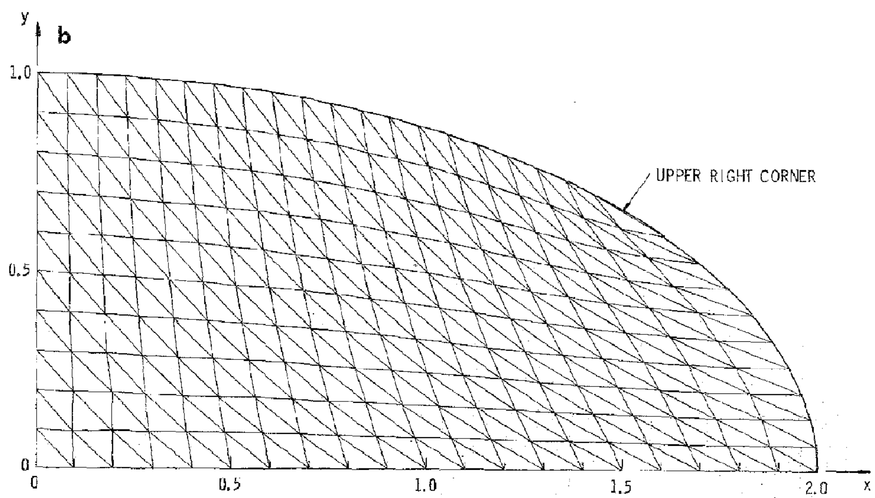

FIG. 3b. Auxiliary mesh in the physical plane of a 1/4-ellipse; $M = 11$, $N = 21$.

The respective maximum local fractional error in the numerical solution was 5.43×10^{-3} and 1.65×10^{-3} at $x = 1.4$, $y = 0$. The convergence rate was thus about 0.304, which is approximately a second order process for which the error bound decreases by a factor of 0.25 when mesh size is halved.

With convergence criteria $E_M < 10^{-9}$, $E_S < 10^{-9}$, the total CP time was 8.03 sec for the 6×11 net, and 18.60 sec for the 11×21 net on a CDC-6600 computer. For mesh generation, the numbers of iterations were 88 and 78 for the initial approximation and 33 and 76 for the final approximation in the above two cases, respectively.

The complete computer program and tables of results for this problem are given in Ref. [3]. The usual SOR process with Eq. (22b) was convergent in this case.

B. *MHD Duct Flow Problem* [19]

The governing equations are

$$\mathscr{L}\phi + F = 0, \qquad F = Ha\,\frac{\partial\psi}{\partial y}, \qquad \mathscr{L} = \nabla^2, \tag{32a, b, c}$$

$$\mathscr{L}\psi + G = 0, \qquad G = 1 + Ha\,\frac{\partial\phi}{\partial y}, \qquad \mathscr{L} = \nabla^2, \tag{33a, b, c}$$

where

ϕ — nondimensional (axial) induced magnetic inductance, (B in Ref. [19])
ψ — nondimensional axial velocity (V in Ref. 19),
Ha — Hartman number (M in Ref. [19]).

The boundary condition is

$$\frac{\partial\phi}{\partial n} + \alpha_3\phi = \alpha_1\frac{\partial\phi}{\partial x} + \alpha_2\frac{\partial\phi}{\partial y} + \alpha_3\phi = 0, \tag{34}$$

$$\alpha_1 = \cos(n, x), \qquad \alpha_2 = \cos(n, y), \tag{34a}$$

and

$$\psi = 0, \tag{35}$$

where n is the outer normal from the fluid and α_3 is the conduction parameter (ϕ_0 in Ref. [19]). The computer program is given in Ref. [20].

As a vector equation with two components at each node, the matrix equation, having large off-diagonal terms, is solved by SOR. However, simple reduction in the underrelaxation factor of the "nonhomogeneous term," F and G, does not assure convergence. For a nonlinearly spaced boundary lattice, the SOR procedure with an underrelaxation factor of 0.025 (Appendix B) with ω given by Eq. (22b) failed to converge, while a direct-and-reverse SOR procedure[8] converged. The latter also converged for the example given below for $\omega = 1$ without underrelaxed terms, at a faster rate [17].

As an example, the results for the case of a perfectly conducting square for a Hartman number of 6 (Fig. 4) were compared with Fourier series solution. The mesh was finer nearer the corners. The maximum fractional error of ϕ_{SOR} with respect to ϕ_{mx} is about 0.024 and that of ψ_{SOR} with respect to ψ_{mx} is about 0.045. The absolute error of the latter is only 1.5×10^{-3}. Higher accuracies can be obtained with finer nets. Due to the special nature (large off-diagonal matrix coefficient) of this set of equations, considerably more machine time is required. With $E_M < 10^{-8}$, $E_S < 10^{-6}$, the CP time for $M = 21$, $N = 21$ was about 5 min on a CDC-6400 computer. The number of iterations to obtain the solution is of the order of 600. More cases are given in Ref. [20].

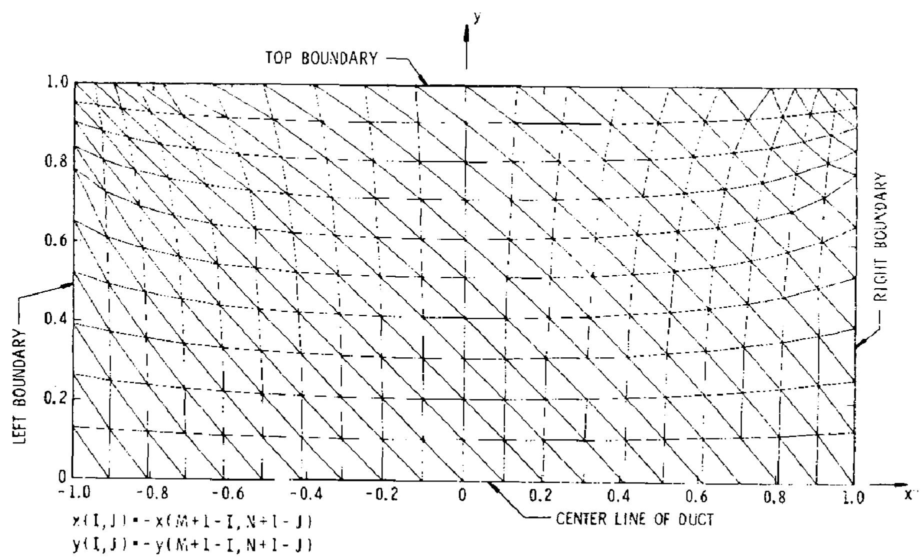

FIG. 4. Auxiliary mesh for a square duct with finer nets near corners; diagonal-symmetric nodes, $M = N = 21$.

V. CONCLUSIONS

The basic idea and the usefulness of a "machine transformation" have been well demonstrated by the examples shown. It seems to possess a high potential to apply

[8] This is a point-by-point SOR procedure, first in direct order, then in reverse order back to the first point to complete one iteration.

directly or seminumerically to many physical problems and partial differential equations (not necessarily purely elliptic). Future research in extending its applications appears to be a significant task for research engineers and mathematicians.

APPENDIX A: COEFFICIENTS OF THE TRANSFORMED EQUATION

The coefficients in Eq. (13) are

$$\bar{A} = A\xi_x^2 + B\xi_x\xi_y + C\xi_y^2, \tag{A-1}$$
$$\bar{B} = A(2\xi_x\eta_x) + B(\xi_y\eta_x + \eta_y\xi_x) + C(2\xi_y\eta_y), \tag{A-2}$$
$$\bar{C} = A\eta_x^2 + B\eta_x\eta_y + C\eta_y^2, \tag{A-3}$$
$$\bar{D} = A\xi_{xx} + B\xi_{xy} + C\xi_{yy} + D\xi_x + E\xi_y, \tag{A-4}$$
$$\bar{E} = A\eta_{xx} + B\eta_{xy} + C\eta_{yy} + D\eta_x + E\eta_y, \tag{A-5}$$
$$\bar{F} = F,$$

where

$$\xi_{xx} = \frac{1}{J}[\xi_x y_{\xi\eta} + \eta_x y_{\eta\eta}] - \frac{y_\eta}{J^2}[\xi_x J_\xi + \eta_x J_\eta], \tag{A-6}$$

$$\xi_{xy} = \frac{1}{J}[\xi_y y_{\xi\eta} + \eta_y y_{\eta\eta}] - \frac{y_\eta}{J^2}[\xi_y J_\xi + \eta_y J_\eta], \tag{A-7}$$

$$\xi_{yy} = -\frac{1}{J}[\xi_y x_\xi + \eta_y x_{\eta\eta}] + \frac{x_\eta}{J^2}[\xi_y J_\xi + \eta_y J_\eta], \tag{A-8}$$

$$J_\xi = y_\eta x_{\xi\xi} + x_\xi y_{\xi\eta} - y_\xi x_{\xi\eta} - x_\eta y_{\xi\xi}, \tag{A-9}$$
$$J_\eta = y_\eta x_{\xi\eta} + x_\xi y_{\eta\eta} - y_\xi x_{\eta\eta} - x_\eta y_{\xi\eta}, \text{ where} \tag{A-10}$$

ξ_x, ξ_y, η_x, η_y, J are given in Eqs. (3) and (5a-d). It can be readily shown that

$$4\bar{A}\bar{C} - \bar{B}^2 = (4AC - B^2)(\xi_x\eta_y - \xi_y\eta_x)^2 = (4AC - B^2).$$

Thus, the equation does not change type.

APPENDIX B: GOVERNING DIFFERENCE EQUATION FOR THE INTERIOR POINTS

$$\mathscr{L}\phi + F = 0 \text{ is}$$

$$\sum_{k=1}^{6} C_k\phi_k - C_0\phi_0 + \bar{F} = R, \qquad R = 0 \text{ at all points,} \tag{B-1}$$

where

$$C_1 = \tfrac{1}{2}\bar{B} + \bar{C} + \tfrac{1}{6}\bar{D} + \tfrac{1}{3}\bar{E}, \tag{B-2}$$

$$C_2 = -\tfrac{1}{2}\bar{B} - \tfrac{1}{6}\bar{D} + \tfrac{1}{6}\bar{E}, \tag{B-3}$$

$$C_3 = \bar{A} + \tfrac{1}{2}\bar{B} - \tfrac{1}{3}\bar{D} - \tfrac{1}{6}\bar{E}, \tag{B-4}$$

$$C_4 = \tfrac{1}{2}\bar{B} + \bar{C} - \tfrac{1}{6}\bar{D} - \tfrac{1}{3}\bar{E}, \tag{B-5}$$

$$C_5 = -\tfrac{1}{2}\bar{B} + \tfrac{1}{6}\bar{D} - \tfrac{1}{6}\bar{E}, \tag{B-6}$$

$$C_6 = \bar{A} + \tfrac{1}{2}\bar{B} + \tfrac{1}{3}\bar{D} + \tfrac{1}{6}\bar{E}, \tag{B-7}$$

$$C_0 = C_8 = 2\bar{A} + \bar{B} + 2\bar{C} = \sum_{k=1}^{6} C_k . \tag{B-8}$$

Acknowledgments

The author would like to thank the SwRI Planning Council for supporting this research under its internal research program and, in particular, Dr. Dana Young, Senior Vice-President, for his serious support and interest. The author would also like to thank Mr. Arthur Muller, Senior Mathematician, for carrying the load of computer programming, Drs. Vivian O'Brien, William L. Ko, and Philip H. Francis for their helpful discussions, Dr. F. T. Dodge for his advice in revising the manuscript, and Mr. Michael Sissung for his technical editing.

References

1. L. M. Milne-Thomson, "Theoretical Hydrodynamics," 2nd ed., MacMillan, New York, 1950.
2. H. Kober, "Dictionary of Conformal Representation," 2nd ed., Dover Publications, New York, 1957.
3. W. H. Chu, "Development of a General Finite Difference Approximation for a General Domain. Part I: Machine Transformation," Final Report IR Project 02-9043, Southwest Research Institute, San Antonio, Texas, 1970.
4. A. M. Winslow, Numerical solution of the quasi-linear Poisson equation in a non-uniform triangular mesh, *J. Computational Phys.* **2** (1966), 149-72.
5. S. K. Godunov and G. P. Prokopov, The calculation of conformal mappings and the construction of difference networks, *U. S. S. R. Computational Math. Math. Phys.* **7** (1967), 5.
6. S. K. Godunov and G. P. Prokopov, The solution of differential equations by the use of curvilinear difference networks, *U. S. S. R. Computational Math. Math. Phys.* **8** (1968), 1.
7. W. H. Chu, Low-gravity fuel sloshing in an arbitrary axisymmetric rigid tank, ASME Paper 70-APM-EEE, *J. Appl. Mech.* **37** (1970), 828–837.
8. R. Courant, Differential and Integral Calculus," Vol. II, Blackie & Son, Ltd., London & Glasgow, 1956, p. 133.
9. W. E. Milne, "Numerical Solution of Differential Equations," John Wiley & Sons, New York, 1953.
10. G. E. Forsythe and W. R. Wasow, "Finite Difference Methods for Partial Differential Equations," John Wiley & Sons, New York, 1960.

11. O. D. Kellogg, "Foundations of Potential Theory," Frederick Ungar, New York, 1929.
12. J. B. Scarborough, "Numerical Mathematical Analysis," 2nd ed., Johns Hopkins Press, Baltimore, Md., 1950.
13. David Young, "Iterative Methods for Solving Partial Difference Equations of the Elliptic Type," *Trans. Amer. Math. Soc.* **76** (1954), 92–111.
14. R. S. Varga, "Matrix Iterative Analysis," Prentice-Hall, Englewood Cliffs, N. J., 1962.
15. J. H. Wilkinson, "The Algebraic Eigenvalue Problem," Oxford University Press, London, 1965.
16. W. H. Chu, Sloshing of an arbitrary two-dimensional tank with flat mean surface (Section 3, Part I, Final Report, Contract No. NAS1-9890, SwRI Project 02-2820) *C.A.S.I. Trans.* **4** (1971), 58–60.
17. W. H. Chu, "A Direct-Reverse Successive Over-Relaxation Procedure," Part III, Final Report, Project IR-02-9043-01, Southwest Research Institute, San Antonio, Texas, 1970.
18. I. W. Sokolnikoff, "Mathematical Theory of Elasticity," 2nd ed., pp. 115, 121, McGraw-Hill, New York, 1956.
19. W. H. Chu, On MHD flow in a rectangular duct of arbitrary conductivity for arbitrary Hartman number, *J. Appl. Mech.* (1969).
20. W. H. Chu, "A Numerical Method for Solving General MHD Duct Flow Problems," Part II, Final Report, Project IR-02-9043-01, Southwest Research Institute, San Antonio, Texas, 1970. To be presented to the ASME 1971 Winter Annual Meeting and to appear in *J. Appl. Mech.*