

# Numerical Solution for Flux Components in Potential Flow\*

By Dale U. von Rosenberg

**Abstract.** Values of the flux components are often desired in potential flow problems. Second-order correct finite-difference analogs are developed for the differential equations defining these flux components. Two iterative methods of solving the resulting finite-difference equations are presented. Experimental results indicate the most efficient value of the iteration parameter and demonstrate that the number of iterations required is approximately proportional to the square root of the number of points in the grid.

**1. Introduction.** Many important physical problems can be described by a potential field. Included in these are the flow of heat, the flow of electricity, and the flow of fluids in porous media. For ideal fluid flow problems a potential is defined only to aid in the solution, and the velocity is the dependent variable of interest. Even in cases where the potential corresponds to a real physical variable, such as heat conduction and flow of fluids in porous media, the flux is often the variable of interest.

For two-dimensional Cartesian co-ordinates, the differential equation which defines the potential is

$$(1) \quad \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0,$$

where  $T$  is the potential,  $y$  is one Cartesian co-ordinate,  $z$  is the other Cartesian co-ordinate.

The flux components can be defined in terms of the potential as

$$(2a) \quad v = -k \frac{\partial T}{\partial y},$$

$$(2b) \quad w = -k \frac{\partial T}{\partial z},$$

where  $v$  is the flux component in the  $y$  direction,  $w$  is the flux component in the  $z$  direction,  $k$  is the transport coefficient.

A great number of potential flow problems can be solved by various analytical techniques. However, a numerical solution is required for many boundary conditions. A number of methods have been developed for numerically solving Eq. (1) for the potential. When values of the flux are desired, the flux components must then be determined from the numerical solution for the potential by finite-difference analogs to Eqs. (2a) and (2b).

---

Received July 15, 1966. Revised February 13, 1967.

\* This work has been supported by NASA Contract NAS8-20136 issued at Marshall Space Flight Center, Huntsville, Alabama.

**2. Equations Defining Flux Components.** The method described in this paper yields a numerical solution directly for the flux components from the defining partial differential equations. Since there are two flux components in the two-dimensional case, two equations are required. The first of these results from the continuity principle and is the equation which yields Eq. (1) in terms of the potential. In terms of the flux components, this equation is

$$(3) \quad \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 .$$

The second of these equations is the irrotationality condition which must hold in order for the potential to be defined by Eq. (2). This relation is

$$(4) \quad \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} = 0 .$$

These equations are completely first order, and they contain the first derivative of each dependent variable with respect to each independent variable.

**3. Boundary Conditions.** Common boundary conditions used in conjunction with Eq. (1) are the specification either of the potential or of the normal derivative of the potential along the boundaries. The most general condition, of course, is specification of a relation between these two along the boundaries. Specification of the potential along a boundary is equivalent to a specification of the tangential flux component along that boundary, while specification of the normal derivative of the potential is equivalent to a specification of the flux component normal to the boundary. The numerical method of solution for the flux components, described herein, has been tested with several types of boundary conditions, including the general type for which a relation between the two flux components is specified.

For purposes of illustrating the numerical method, the boundary conditions used are

$$(5a) \quad w(z, 1) = f(z) ,$$

$$(5b) \quad w(0, y) = g(y) ,$$

$$(5c) \quad v(z, 0) = p(z) ,$$

$$(5d) \quad v(1, y) = q(y) .$$

These boundary conditions are equivalent to a specification of the potential along the adjacent sides for  $y = 1$  and  $z = 1$  and of the normal derivative of the potential along the other two adjacent sides where  $y = 0$  and  $z = 0$ .

**4. Finite-Difference Equations.** A set of grid points with equal increments in the two directions is imposed on the region. This grid is illustrated in Fig. 1. Indices are used to designate location in the grid. These indices are defined by

$$(6a) \quad z_i = (i - 1)\Delta z \quad \text{for } 1 \leq i \leq R ,$$

$$(6b) \quad y_j = (j - 1)\Delta y \quad \text{for } 1 \leq j \leq S ,$$

$$(6c) \quad \text{with } \Delta z = \Delta y .$$

For the case of a square region,  $R = S$ , but these limits will be used as given in Eq. (6) so that the method can be applied directly to a rectangular region. Two subscripts are used with values of the dependent variables. These are defined as

$$(7a) \quad w_{i,j} = w(z_i, y_j),$$

$$(7b) \quad v_{i,j} = v(z_i, y_j).$$

The boundary conditions of Eq. (5) are shown in Fig. 1 in terms of the discrete variables.

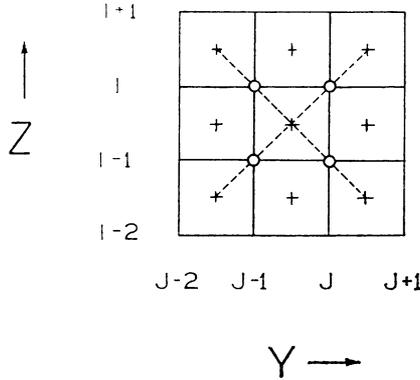


FIGURE 1. Finite-difference grid

The first derivatives of Eqs. (3) and (4) are replaced by centered differences. These differences are written about the point  $z_{i-1/2}, y_{j-1/2}$  which is designated by the cross (+) in Fig. 1. However, only values of the dependent variables on the grid points are used in the finite differences. This method of writing the finite differences has been described previously for equations describing transient, countercurrent flow problems [1]. The analogs for the derivatives of  $v$  are

$$(8a) \quad \frac{\partial v}{\partial z} \simeq \frac{1}{2} \left[ \frac{v_{i,j} - v_{i-1,j}}{\Delta z} + \frac{v_{i,j-1} - v_{i-1,j-1}}{\Delta z} \right],$$

$$(8b) \quad \frac{\partial v}{\partial y} \simeq \frac{1}{2} \left[ \frac{v_{i,j} - v_{i,j-1}}{\Delta y} + \frac{v_{i-1,j} - v_{i-1,j-1}}{\Delta y} \right].$$

Those for  $w$  are similar. These analogs are second-order correct. The truncation will be discussed further in the next section.

Two finite-difference equations can be written for each square of four points in the grid. After the space increments and the factor 1/2 have been eliminated, these equations become

$$(9) \quad w_{i,j} - w_{i-1,j} + w_{i,j-1} - w_{i-1,j-1} + v_{i,j} - v_{i,j-1} + v_{i-1,j} - v_{i-1,j-1} = 0,$$

$$(10) \quad w_{i,j} - w_{i,j-1} + w_{i-1,j} - w_{i-1,j-1} - v_{i,j} + v_{i-1,j} - v_{i,j-1} + v_{i-1,j-1} = 0.$$

Each equation contains eight values of the dependent variables, two at each of the four points. These equations, together with the boundary conditions, define the values of  $w$  and  $v$  at all points in the grid. Since the boundary conditions are specified on opposite boundaries in both directions, a simultaneous solution of all the equa-

tions is necessary. Two iterative methods of effecting this solution are presented in later sections.

**5. Truncation Error.** The complete expressions for the first derivatives are obtained from a Taylor series in two independent variables. The truncation error is obtained by substituting these expressions into the original differential equations. The expressions for the errors can be simplified by use of relations obtained from repeated differentiation of Eqs. (3) and (4). The truncation error for the finite-difference analog to Eq. (3) is

$$(11) \quad E_3 = -2 \sum_{n=0}^{\infty} \left(\frac{\Delta z}{2}\right)^{4n+2} \left(\frac{\partial^{4n+3} w}{\partial z^{4n+3}}\right) \sum_{r=0}^{2n+1} (-1)^r \frac{1}{(2r)!(4n - 2r + 3)!}$$

and that for Eq. (4) is

$$(12) \quad E_4 = -2 \sum_{n=0}^{\infty} \left(\frac{\Delta z}{2}\right)^{4n+2} \left(\frac{\partial^{4n+3} v}{\partial z^{4n+3}}\right) \sum_{r=0}^{2n+1} (-1)^r \frac{1}{(2r)!(4n - 2r + 3)!}.$$

**6. Corresponding Difference Equation in Potential.** A finite-difference equation in terms of the potential can be obtained from the finite-difference analog to Eq. (3) and to analogs to Eq. (2). Second-order correct centered difference analogs to Eq. (2) are used to obtain the flux components from the potential. The location of the points used in these analogs are shown in Fig. 2. In order for the flux components at the intersections of the grid lines to be determined from values of the potential by these centered analogs, the potential must be known at the points denoted by the crosses (+). A typical equation for determining a flux component is

$$(13) \quad v_{i,j} = -k \left(\frac{1}{2}\right) \left(\frac{T_{i+1/2,j+1/2} - T_{i+1/2,j-1/2}}{\Delta y} + \frac{T_{i-1/2,j+1/2} - T_{i-1/2,j-1/2}}{\Delta y}\right).$$

When such analogs for the flux components at the points denoted by circles (O) in Fig. 2 are substituted into the finite-difference equation for Eq. (3), the finite-

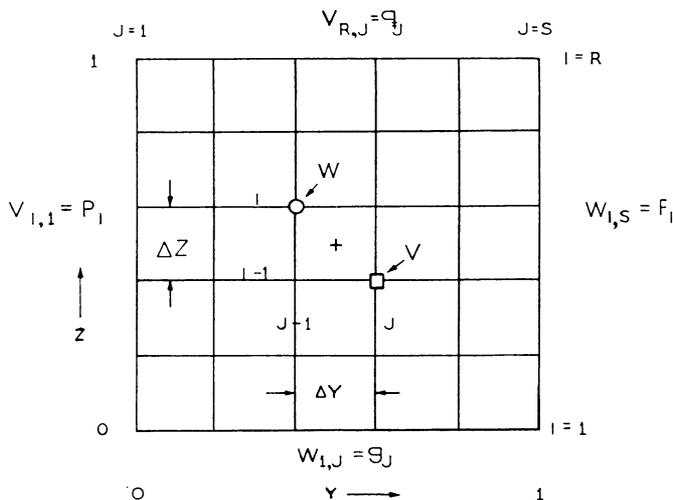


FIGURE 2. Location of points for potential

difference equation for the potential is obtained. This is

$$(14) \quad \frac{T_{i+1/2,j+1/2} + T_{i+1/2,j-3/2} + T_{i-3/2,j+1/2} + T_{i-3/2,j-3/2} - 4T_{i-1/2,j-1/2}}{2(\Delta y)^2} = 0.$$

This equation is the familiar five-point analog to Eq. (1) with the grid lines rotated  $45^\circ$  from those used to define the flux components, as shown in Fig. 2. Furthermore, the increment size is  $\sqrt{2}$  times that used in the grid for the flux. When analogs similar to Eq. (13) are substituted into the finite-difference analog to Eq. (4), the irrotationality condition, all terms cancel as they should.

**7. Nature of Iterative Methods of Solution.** As mentioned previously, two iterative methods of solving Eqs. (9) and (10) will be described. The first method effects the simultaneous solution of the finite-difference equations from a single row in the grid. In the second method, the equations from two adjacent rows in the grid are solved simultaneously. The methods can be formulated and carried out either in the  $y$ -direction or in the  $z$ -direction. In fact, the convergence is more rapid if the iteration is carried out alternately in the  $y$ -direction and then in the  $z$ -direction. However, iteration in only one direction is convergent. In the presentation of the methods in this paper, the solution is given for one or two rows of equations in the  $y$ -direction.

The iteration procedure is begun at a boundary, where the values of one of the flux components are known for a whole row of points. The other flux component is unknown along this row, as are both components on the second row of points. In the equations presented below, the iteration is begun at the  $z = 0$  boundary, where  $w$  is given by the boundary condition. See Fig. 1. The finite-difference equations, Eqs. (9) and (10), relate the dependent variables along these two rows of points. However, there are three rows of unknowns and only two rows of equations. It is necessary, therefore, to assume values of  $v$  along the second row of points. Values of  $v$  on the first row of points and  $w$  on the second row are then computed from Eqs. (9) and (10) based on this assumption. These computed values of  $w$  are then used with assumed values for  $v$  on the third row to compute values of  $w$  on the third row and  $v$  on the second row. This procedure is continued across the region. For the last row of equations, however, the values of  $v$  do not need to be assumed, since they are given by the boundary condition, as shown in Fig. 1, for  $z = 1$ . The conditions on each boundary are thus introduced once on each sweep across the region. On the next sweep of the region, the values obtained in the first sweep are used in place of the assumed values.

In order to increase the rate of convergence, an iteration parameter,  $\epsilon$ , is introduced into the Eqs. (9) and (10). These equations become

$$(9a) \quad w_{i,j}^{(m+1)} - w_{i-1,j}^{(m+1)} + w_{i,j-1}^{(m+1)} - w_{i-1,j-1}^{(m+1)} + v_{i,j}^{(m)} - v_{i,j-1}^{(m)} + v_{i-1,j}^{(m+1)} - v_{i-1,j-1}^{(m+1)} \\ = \epsilon [(w_{i,j-1}^{(m)} - w_{i,j-1}^{(m+1)}) + (v_{i-1,j}^{(m)} - v_{i-1,j}^{(m+1)})];$$

$$(10a) \quad w_{i,j}^{(m+1)} - w_{i,j-1}^{(m+1)} + w_{i-1,j}^{(m+1)} - w_{i-1,j-1}^{(m+1)} - v_{i,j}^{(m)} + v_{i-1,j}^{(m+1)} - v_{i,j-1}^{(m)} + v_{i-1,j-1}^{(m+1)} \\ = \epsilon [-(w_{i,j-1}^{(m)} - w_{i,j-1}^{(m+1)}) + (v_{i-1,j}^{(m)} - v_{i-1,j}^{(m+1)})].$$

The superscripts denote the level of the iterate. On the left side of the equations, only two values at the old iterate (denoted by  $m$ ) are used. These are the values of  $v$

on the  $i$ th row which were assumed for the initial sweep. The values of  $w$  on the  $(i - 1)$ th row are known at the new iteration level from the computations on the previous row of equations. The values of  $w$  on the  $i$ th row and  $v$  on the  $(i - 1)$ th row are to be computed from the simultaneous solution of this row of equations. The iteration parameter is introduced into the right sides of the equations as a coefficient of the differences between the old and the new values of one  $v$  and one  $w$ .

The values of  $v$  and  $w$  used in this iteration term are not located at the same point in the grid. Reference to Fig. 1 will show that the  $v$  appearing in this term is located at the point designated by the square ( $\square$ ) and the  $w$  at the point designated by the circle ( $\circ$ ). The location of the  $v$  is the one point of the four in the square which is farthest from the boundary conditions specifying  $v$  in both the  $y$ - and  $z$ -directions. The  $w$  point is similarly located.

**8. Single-Row or Point-Wise Iterative Method.** The method in which a single row of equations is solved simultaneously is the simpler of the two. Because of the nature of the equations, the method is actually a point-wise or explicit method. The first step in developing this method is to add Eqs. (9a) and (10a). This resulting equation contains only two of the four values to be computed. When multiplied by one-half and written for  $j$ , it is

$$(15) \quad w_{i,j}^{(m+1)} + (1 + \epsilon)v_{i-1,j}^{(m+1)} = w_{i-1,j-1}^{(m+1)} + v_{i,j-1}^{(m)} + \epsilon w_{i-1,j}^{(m)}.$$

Another equation containing the same two unknowns can be obtained from one-half the difference between these equations written for  $j + 1$ . This equation is

$$(16) \quad (1 + \epsilon)w_{i,j}^{(m+1)} - v_{i-1,j}^{(m+1)} = w_{i-1,j+1}^{(m+1)} - v_{i,j+1}^{(m)} + \epsilon w_{i,j}^{(m)}.$$

These equations can be solved simultaneously to yield, for  $2 \leq j \leq (S - 1)$ ,

$$(17) \quad v_{i-1,j}^{(m+1)} = \frac{(1 + \epsilon)A - B}{1 + (1 + \epsilon)^2},$$

$$(18) \quad w_{i,j}^{(m+1)} = \frac{A + (1 + \epsilon)B}{1 + (1 + \epsilon)^2},$$

where  $A = w_{i-1,j-1}^{(m+1)} + v_{i,j-1}^{(m)} + \epsilon w_{i-1,j}^{(m)}$ ,  $B = w_{i-1,j+1}^{(m+1)} - v_{i,j+1}^{(m)} + \epsilon w_{i,j}^{(m)}$ .

The boundary conditions specify  $v_{i-1,1} = p_{i-1}$  and  $w_{i,S} = f_i$ . The value of  $w$  at the left boundary can be obtained from Eq. (16) as

$$(19) \quad w_{i,1}^{(m+1)} = (p_{i-1} + w_{i-1,2}^{(m+1)} - v_{i,2}^{(m)} + w_{i,1}^{(m)}) / (1 + \epsilon).$$

Likewise, the value of  $v$  at the right boundary can be obtained from Eq. (15) as

$$(20) \quad v_{i-1,S}^{(m+1)} = (-f_i + w_{i-1,S-1}^{(m+1)} + v_{i,S-1}^{(m)} + v_{i-1,S}^{(m)}) / (1 + \epsilon).$$

Eqs. (17)–(20) provide the relations necessary for computing the flux components by the single-row iterative method.

**9. Double-Row Iterative Method.** The double-row method utilizes the explicit nature of the equations of the single-row method. The values of  $w$  on the  $i$ th row are expressed in terms of the values of  $w$  on the  $(i - 1)$ th row and  $v$  on the  $i$ th row by Eqs. (18) and (19) with  $\epsilon = 0$ . These are substituted into Eqs. (15) and (16) written between the  $i$ th and  $(i + 1)$ th rows. The resulting set of equations can be solved to

yield values of  $w$  on the  $(i + 1)$ th row and values of  $v$  on the  $i$ th row. The solution of these equations requires values of  $v$  at the old iterate on the  $(i + 1)$ th row and values of  $w$  at the new iterate on the  $(i - 1)$ th row. Values of  $w$  on the  $i$ th row can then be computed from the values of  $v$  at the new iterate on the  $i$ th row by Eqs. (18) and (19) with  $\epsilon = 0$ . Likewise, the values of  $v$  on the  $(i - 1)$ th row can be computed from these values of  $v$  on the  $i$ th row by Eqs. (17) and (20) with  $\epsilon = 0$ . In this manner, values of  $v$  at the old iterate are required on only every other row of points for each sweep across the region. This method, consequently, converges more rapidly than does the single-row method.

The direct solution of these equations has been effected by separating the complete system into two bi-tridiagonal systems of equations. The general equations of each system are the same, and they are

$$(21) \quad -\epsilon w_{i+1,j}^{(m+1)} - \frac{1}{2}v_{i,j-2}^{(m+1)} + (3 + \epsilon)v_{i,j}^{(m+1)} - \frac{1}{2}v_{i,j+2}^{(m+1)} \\ = \frac{1}{2}[w_{i-1,j-2}^{(m+1)} - w_{i-1,j+2}^{(m+1)}] + v_{i+1,j-1}^{(m)} + v_{i+1,j+1}^{(m)} + \epsilon[v_{i,j}^{(m)} - w_{i+1,j}^{(m)}];$$

$$(22) \quad -(1 + \epsilon)w_{i+1,j-2}^{(m+1)} + 3w_{i+1,j}^{(m+1)} + (4 + 3\epsilon)v_{i,j}^{(m+1)} \\ = w_{i-1,j-2}^{(m+1)} + w_{i-1,j}^{(m+1)} + 4v_{i+1,j-1}^{(m)} + \epsilon[3v_{i,j}^{(m)} - w_{i+1,j-2}^{(m)}].$$

In one system the  $j$  index takes on odd values, and in the other system it takes on even values. The boundary equations of each system are obtained from suitable combinations of the original equations.

A number of solution algorithms have been developed for these systems of equations, but all of these develop significant round-off error for grids of 20 points in each direction. Work is continuing in an effort to find satisfactory algorithms.

**10. Experimental Study of Convergence Rate.** A number of runs were made on an IBM 7044 computer to study the number of iterations required for convergence. The boundary conditions of the test problem are

$$(23a) \quad w(z, 1) = z,$$

$$(23b) \quad w(0, y) = 0,$$

$$(23c) \quad v(z, 0) = 0,$$

$$(23d) \quad v(1, y) = -y.$$

The solution to Eqs. (3) and (4) with these boundary conditions is  $w = z$  and  $v = -y$  for all  $y$  and  $z$ . Since all derivatives above the first are zero for this problem, the numerical solution will converge to the analytic solution for all grid sizes. Furthermore, this solution is a particularly easy one to check for convergence. The initial guess used in all the test runs was  $v = w = 0$  at all points in the grid.

The first purpose of this study was to determine experimentally the most efficient value of the iteration parameter,  $\epsilon$ . The second was to compare convergence rates for the double-row method and for the single-row method. The third purpose was to compare the effect of grid size on the number of iterations required for convergence.

The runs made to determine the most efficient parameter were made on a square

grid with ten increments on each side. Consequently, there were 200 values of the dependent variables to be determined. The method diverged for the two negative values of the parameter tested, and it converged for all positive values and for zero.

For  $\epsilon = 0$ , the values of the dependent variables approached the correct values asymptotically from the initial guess of zero. For the three positive values of  $\epsilon$  tested, namely,  $\frac{1}{2}$ , 1, and 2, the intermediate values of the dependent variables, in some parts of the grid, increased above the correct values and then converged to those values by a damped oscillation. Furthermore, convergence was more rapid for all three of these values than for  $\epsilon = 0$ . Of the three positive values,  $\epsilon = 1$  was the most efficient. At the end of 20 double iteration steps (one in the  $y$ -direction and one in the  $z$ -direction) no value of the dependent variables differed from the correct value by more than two in the fourth place. Most values were closer than this. For the other two values of  $\epsilon$ , some values of the dependent variables differed in the third place after 20 steps.

The double-row method converged in approximately half as many iterations as the single-row method. For  $\epsilon = 1$  and a  $10 \times 10$  grid, the values obtained after ten double steps by the double-row method were approximately the same as those obtained after 20 double steps by the single-row method. After 20 double steps by the double-row method, the values were almost completely converged to six places. Only nine of the 200 values differed from the correct values by more than three in the sixth place. The largest difference was seven in the sixth place.

The number of iterations required for convergence for a square grid is approximately proportional to the number of points along one side of the square or, consequently, to the square root of the total number of points in the grid. For a square grid with 100 points on a side, 200 double steps were required to obtain the same extent of convergence as was obtained by 20 double steps with a  $10 \times 10$  grid. In both of these test runs, the single-row method was used with  $\epsilon = 1$ . Similar results were obtained for the double-row method with  $\epsilon = 0$  for square grids with 10 and 20 points on each side.

**11. Experimental Study of Truncation Error.** A study of the truncation error was made for flow near a unit source at the origin. The exact solution for this problem is given by

$$(24) \quad w = z/(y^2 + z^2),$$

$$(25) \quad v = y/(y^2 + z^2).$$

The flux components were computed in a square region with boundaries at  $y = 1/2$ ,  $z = 1/2$ ,  $y = 39/2$ , and  $z = 39/2$  for increment sizes of 1,  $1/2$ , and  $1/4$ ; and the resulting values were compared with the exact solution. The truncation error was approximately 30%, 7%, and 1.5% for the three grid sizes; this variation is in line with the second-order correct nature of the finite-difference analogs. The single-row method of solution was used, and round-off error was negligible even for the largest grid of 77 points per side.

**12. Comparison with Alternating-Direction-Implicit Method for Potential.** The solution for flux components was compared with the alternating-direction-implicit

method of solution for the potential in a square region for equivalent boundary conditions. The boundary conditions in terms of potential are

$$(26a) \quad \text{at } y = 0, \partial T / \partial y = 0 \quad \text{for all } z ,$$

$$(26b) \quad \text{at } z = 0, \partial T / \partial z = 0 \quad \text{for all } y ,$$

$$(26c) \quad \text{at } y = 1, T = 0 \quad \text{for } z < 1 ,$$

$$(26d) \quad \text{at } z = 1, T = 1 \quad \text{for } y < 1 .$$

Equivalent boundary conditions in terms of flux components are

$$(27a) \quad \text{at } y = 0, v = 0 \quad \text{for all } z ,$$

$$(27b) \quad \text{at } z = 0, w = 0 \quad \text{for all } y ,$$

$$(27c) \quad \text{at } y = 1, w = 0 \quad \text{for } z < 1 ,$$

$$(27d) \quad \text{at } z = 1, v = 0 \quad \text{for } y < 1 ,$$

$$(27e) \quad \text{at } y = 1 \text{ and } z = 1, v = 1 \text{ and } w = -1 .$$

Numerical solutions were obtained for a grid of 20 increments per side; thus, there were 400 points at which either the potential or the flux components were to be obtained. The initial iterates for each method were essentially equivalent.

For the alternating-direction-implicit method the set of nine iteration parameters which result in most rapid convergence was used. This set of parameters is given by Young [2]. Convergence was obtained in two cycles of the parameters or in 18 iterations.

No analysis has been made to obtain a set of iteration parameters for most rapid convergence in the solution for the flux components. Consequently, this solution was effected using a value of unity for the parameter. Convergence was obtained in 30 iterations when the double-row method of solution was used. This method does not compare unfavorably with the alternating-direction-implicit method, and the use of a set of more efficient iteration parameters for the flux component method should decrease the amount of iterations required.

**13. Conclusion.** An efficient numerical method for the determination of the flux components in potential flow has been developed. Two iterative methods for solving the resulting finite-difference equations are described. Experimental results which determine the most efficient value of the iteration parameter and evaluate the relative efficiencies of the two iteration techniques are presented. These results also show that the number of iterations required for convergence is approximately proportional to the square root of the number of points in the grid.

Tulane University  
New Orleans, Louisiana

1. E. H. Herron & D. U. VON ROSENBERG, "An efficient numerical method for the solution of pure convective transport problems with split boundary conditions," *Chem. Eng. Sci.*, v. 21, 1966, p. 337.

2. D. YOUNG, *Modern Mathematics for the Engineer*, Second Series, McGraw-Hill, New York, 1961, p. 393. MR 23 #B2205.