COMPUTER SCIENCES DEPARTMENT
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin

# FAST FINITE-DIFFERENCE SOLUTION OF BIHARMONIC PROBLEMS

by

D. Greenspan* and D. Schultz**

Technical Report #96

August 1970

*Computing Center and Department of Computer Sciences, Univ. of Wisconsin, Madison.

**Department of Mathematics, Univ. of Wisconsin, Milwaukee.

# FAST FINITE-DIFFERENCE SOLUTION OF BIHARMONIC PROBLEMS

## 1. Introduction

The advent of the high speed digital computer has been conducive to the development of new, and renewed interest in old, numerical techniques for biharmonic problems (see, e.g., references [1]-[4], [7]-[15]). Of these, the direct finite-difference method has been restrictive in that iteration for the resulting linear algebraic system is inherently slow [10]. Our objective here is to describe a finite-difference method for biharmonic problems which utilizes the fast iterative methods available for harmonic problems. The biharmonic then will be treated as a system of second order elliptic equations. However, as will be shown, it will be necessary, in addition, to incorporate a simple smoothing process to assure convergence [6]. After the method is described and illustrated, it will be shown how to adapt it easily to crack-type boundary value problems [2], [13].

## 2. A Prototype Problem

For simplicity, we will consider first a prototype biharmonic problem. Let $S$ be a unit square and let $R$ be the interior of $S$. Consider the boundary value problem on $R \cup S$ in which one seeks a function $u(x,y)$ which is continuous on $R \cup S$, which satisfies the biharmonic equation on $R$, and which satisfies prescribed function and normal derivative conditions on S . More precisely, let S have vertices A(0,0), B(1,0), C(1,1) and D(0,1), as shown in Figure 2.1. Let $f_i$

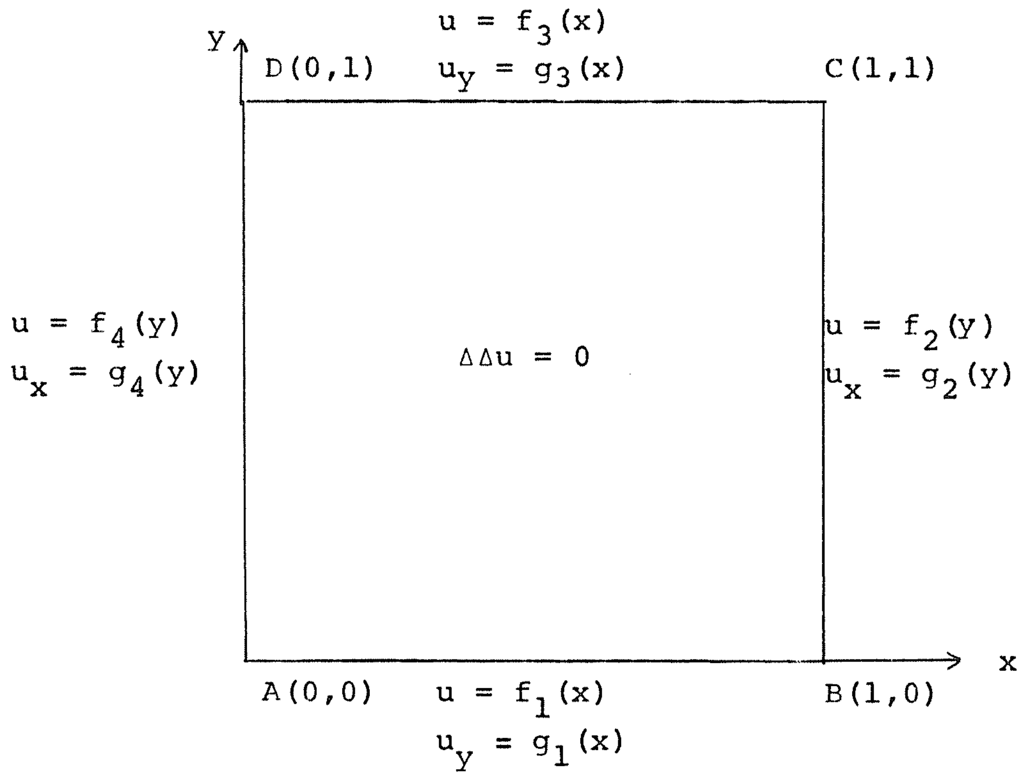


Figure 2.1

and $g_i$, $i = 1,2,3,4$, be continuous functions of a single variable. Then our problem is to find a function $u(x,y)$ which is continuous on $R \cup S$, which is a solution on $R$ of the biharmonic equation.

$$\Delta\Delta u = 0 \tag{2.1}$$

and which satisfies the boundary conditions

$$u(x,0) = f_1(x), \quad u_y(x,0) = g_1(x); \quad 0 \le x \le 1$$
$$u(1,y) = f_2(y), \quad u_x(1,y) = g_2(y); \quad 0 \le y \le 1$$
$$u(x,1) = f_3(x), \quad u_y(x,1) = g_3(x); \quad 0 \le x \le 1$$
$$u(0,y) = f_4(y), \quad u_x(0,y) = g_4(y); \quad 0 \le y \le 1 ,$$

as shown in Figure 2.1.

## 3. The Numerical Method

The method to be developed for prototype problem (2.1)-(2.2) can be described in general as follows. First set

(3.1) $$\Delta u = -\omega$$

on $R$, so that from (2.1)

(3.2) $$\Delta \omega = 0.$$

Consideration of (2.1) will be replaced by consideration of the system (3.1)-(3.2). Physically, one can interpret $u$ and $\omega$, from, say, the fluid dynamics point of view, as stream and vorticity functions, respectively. Next, replace $R$ and $S$ by sets of grid points $R_h$ and $S_h$ in the usual way [5]. We will construct on $R_h$ a sequence of discrete functions

(3.3) $$u^{(0)}, u^{(1)}, u^{(2)}, u^{(3)}, \ldots .$$

and on $R_h \cup S_h$ a sequence of discrete functions

(3.4) $$\omega^{(0)}, \omega^{(1)}, \omega^{(2)}, \omega^{(3)}, \ldots$$

which, with the aid of an elementary smoothing procedure, will converge to yield the numerical solution. The functions $u^{(k)}$

of (3.3) will result from a discretization of (3.1) while the functions $\omega^{(k)}$ of (3.4) will result from a discretization of (3.2). The double sequences (3.3) and (3.4) will be generated in an alternating fashion as follows:

$$\begin{array}{ccccccccc} u^{(0)} & & u^{(1)} & & u^{(2)} & & u^{(3)} & & \\ & \nearrow & \downarrow & \nearrow & \downarrow & \nearrow & & & \cdot \;\cdot \;\cdot \;\cdot \\ \omega^{(0)} & & \omega^{(1)} & & \omega^{(2)} & & & & \end{array}$$

Let us then now give a precise formulation of the algorithm:

Step 1.

For $n$ a positive integer, let $h = \frac{1}{n}$ be the grid size, and construct and number in the usual way the interior grid points $R_h$ and the boundary grid points $S_h$.

Step 2.

Set
$$u^{(0)} = 0 \;, \quad \text{on } R_h$$
$$\omega^{(0)} = 0 \;, \quad \text{on } R_h \cup S_h \,.$$

Step 3.

To produce the second iterate $u^{(1)}$ of (3.3) on $R_h$ proceed as follows. At each point of $R_h$ write down the difference analogue

$$(3.5) \qquad -4u(x,y) + u(x+h,y) + u(x,y+h) + u(x-h,y) + u(x,y-h) = -h^2\,\omega^{(0)}(x,y)$$

of (3.1) and, whenever possible, insert the given boundary values $f_i$ from (2.2). Solve the resulting linear algebraic system by

SOR with over-relaxation factor $r_1$ and denote the solution by $\bar{u}^{(1)}$. Finally, define $u^{(1)}$ on $R_h$ by the smoothing formula

$$u^{(1)} = \rho u^{(0)} + (1-\rho)\bar{u}^{(1)}, \quad 0 \le \rho \le 1 . \tag{3.6}$$

Step 4.

To produce the second iterate $\omega^{(1)}$ of (3.4) on $R_h \cup S_h$ proceed as follows. At each point of $S_h$ of the form (ih,0), i = 0,1,2,...,n, set ([5], pp. 127-129)

$$\bar{\omega}^{(1)}(ih,0) = \frac{4}{h^2} f_1(ih) - \frac{1}{h^2} f_1(ih+h) - \frac{2}{h^2} u^{(1)}(ih,h) - \frac{1}{h^2} f_1(ih-h) + \frac{2}{h} g_1(ih) ; \tag{3.7}$$

at each point of $S_h$ of the form (1,ih), i = 1,2,...,n-1, set

$$\bar{\omega}^{(1)}(1,ih) = \frac{4}{h^2} f_2(ih) - \frac{1}{h^2} f_2(ih+h) - \frac{2}{h^2} u^{(1)}(1-h,ih) - \frac{1}{h^2} f_2(ih-h) - \frac{2}{h} g_2(ih) ; \tag{3.8}$$

at each point of $S_h$ of the form (ih,1), i = 0,1,2,...,n, set

$$\bar{\omega}^{(1)}(ih,1) = \frac{4}{h^2} f_3(ih) - \frac{1}{h^2} f_3(ih+h) - \frac{1}{h^2} f_3(ih-h) - \frac{2}{h^2} u(ih,1-h) - \frac{2}{h} g_3(ih) ; \tag{3.9}$$

and at each point of $S_h$ of the form (0,ih), i = 1,2,...,n-1, set

$$(3.10)\qquad \bar{\omega}(0,ih) = \frac{4}{h^2} f_4(ih) - \frac{2}{h^2} u^{(1)}(h,ih) - \frac{1}{h^2} f_4(ih+h) - \frac{1}{h^2} f_4(ih-h) + \frac{2}{h} g_4(ih) .$$

Next, at each point of $R_h$ write down the difference analogue

$$(3.11)\qquad -4\omega(x,y) + \omega(x+h,y) + \omega(x,y+h) + \omega(x-h,y) + \omega(x,y-h) = 0$$

of (3.2), inserting wherever possible the boundary values obtained from (3.7)-(3.10). Solve the linear algebra system generated by (3.11) by SOR with over-relaxation factor $r_2$ and denote the solution by $\bar{\omega}^{(1)}$. Finally, on all of $R_h \cup S_h$, define $\omega^{(1)}$ by the smoothing formula

$$(3.12)\qquad \omega^{(1)} = \mu\omega^{(0)} + (1 - \mu)\bar{\omega}^{(1)},\ 0 \le \mu \le 1.$$

Step 5.

Proceed next on $R_h$ to determine $u^{(2)}$ from $\omega^{(1)}$ in the same fashion as $u^{(1)}$ was determined from $\omega^{(0)}$. Then on $R_h \cup S_h$ construct $\omega^{(2)}$ from $u^{(2)}$ just as $\omega^{(1)}$ was determined from $u^{(1)}$. In the indicated fashion construct sequences (3.3) and (3.4).

Step 6.

Given $\varepsilon_1 > 0$ and $\varepsilon_2 > 0$, terminate the iteration of Step 5 when both the following are valid:

$$(3.13)\qquad |u^{(k)} - u^{(k+1)}| < \varepsilon_1 \text{ uniformly on } R_h$$

$$(3.14)\qquad |\omega^{(k)} - \omega^{(k+1)}| < \varepsilon_2 \text{ uniformly on } R_h \cup S_h ,$$

and let $u^{(k+1)}$ be the approximate solution on $R_h$ of boundary value problem (2.1)-(2.2).

## 4. Illustrative Example

Typical of the examples run on the UNIVAC 1108 is the following. For the boundary data $f_1 = x^3$, $f_2 = 2y + 1 - 3y^2$, $f_3 = x^3 + 2x - 3$, $f_4 = -3y^2$, $g_1 = 2x$, $g_2 = 2y + 3$, $g_3 = 2x - 6$, and $g_4 = 2y$, the method of Section 3 was executed with $h = 0.05$, $\rho = 0.2$, $u = 0.85$, $\varepsilon_1 = 10^{-4}$, $\varepsilon_2 = 10^{-3}$, $r_1 = 1.8$, $r_2 = 1.0$. Convergence criteria (3.13) and (3.14) were satisfied at $k = 93$ and the numerical solution agreed with the exact solution, $u = x^3 - 3y^2 + 2xy$, to three decimal places. The total running time was under 6 minutes. Deletion of smoothing, that is, setting $\rho = \mu = 0$, in this example resulted in divergence in three outer iterations.

## 5. Extension of the Numerical Method

The method of Section 3 generalizes easily to rectangular type regions in any number of dimensions [5]. Let us then proceed in a different direction and show how to extend the method to a difficult problem of applied interest in which S need not be a simple closed curve. The problem is that of determining an Airy stress function u when one has reentrant boundaries [2], and it can be formulated in general as follows.

Consider the rectangular type plate O A B C D E F A, shown in Figure 5.1, where O is the geometric center of the figure

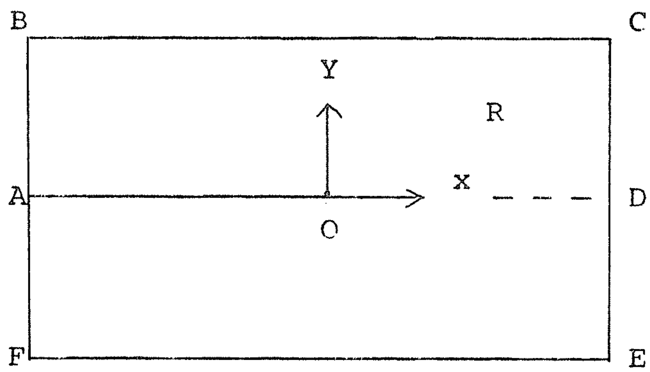


Figure 5.1

and where the boundary sections are parallel to the axes. For convenience, let S represent the connected segments O A B C D E F A, and let R be the finite open region bounded by S . Then if u and its normal derivatives are prescribed in S , one must find u(x,y) which is continuous on $R \cup S$, satisfies the prescribed boundary conditions on S, and satisfies (2.1) on R .

The fundamental difficulty with the above problem is that the point O is a singular point in the sense that higher-order derivatives of u become unbounded in any neighborhood of O . The standard finite-difference methods still work, but the ultrasimplictic convergence proofs based on Taylor expansions are no longer valid. Also, one may have to exert special effort to obtain a high accuracy in a neighborhood of O . In [2], the form of such an effort was based in special series expansions developed by Williams [14]. Here, we shall replace five-point difference equations by nine point approximations at selected points near O . Let us then consider a particular problem and describe the method precisely.

As in [2], consider a problem which is symmetric about AD, so that attention need be restricted only to O A B C D O. Set AB = 0.7, BC = 0.8, OA = 0.4 and fix the boundary conditions as

(5.1) $u = 0$ , $u_y = 0$ ; on OA

(5.2) $u = 0$ , $u_x = 0$ ; on AB

(5.3) $u = 5000x^2$ , $u_y = 0$ ; on BC

(5.4) $u = 3200$ , $u_x = 8000$ ; on CD .

For the boundary value problem defined by (2.1) and (5.1)-(5.4), the method of Section 3 was modified only in the following ways. At grid points $(h,0)$, $(h,h)$, $(0,h)$ and $(-h,h)$ about O , the nine point difference analogue of the Laplace operator [1] was used. At grid points between O and D which were different from $(h,0)$, the five point analogue was used. Symmetry of the solution about AD was incorporated into all formulas. For the parameter choices $h = 0.1$, $\rho = 0.2$, $\mu = 0.85$, $r_1 = 1.8$, $r_2 = 1.0$, $\varepsilon_1 = 0.003$, and $\varepsilon_2 = 2.0$, the iteration converged at $k = 251$ with a running time of two minutes. For $h = 0.05$, $\rho = 0.2$, $\mu = 0.85$, $r_1 = 1.8$, $r_2 = 1.0$, $\varepsilon_1 = .001$, and $\varepsilon_2 = 2.0$ , the iteration converged at $k = 248$ with a running time of 5 minutes. For $h = \frac{1}{80}$, $\rho = 0.2$, $\mu = 0.85$, $r_1 = 1.8$, $r_2 = 1.0$,

$\varepsilon_1 = .01$, $\varepsilon_2 = 20$, and with the previous results with linearly interpolated values used for $u^{(0)}$ and $\omega^{(0)}$, the iteration converged at $k = 196$ with a running time of 15 minutes. Shown in the Table are typical results from the case $h = \frac{1}{80}$ .

TABLE

| x | y | u | x | y | u | x | y | u |
|---|---|---|---|---|---|---|---|---|
| 0.1 | 0 | 481 | 0.2 | 0.2 | 1500 | 0.3 | 0.4 | 2417 |
| 0.2 | 0 | 1333 | 0.3 | 0.2 | 2351 | -0.3 | 0.5 | 39 |
| 0.3 | 0 | 2299 | -0.3 | 0.3 | 11 | -0.2 | 0.5 | 162 |
| -0.3 | 0.1 | -4 | -0.2 | 0.3 | 66 | -0.1 | 0.5 | 383 |
| -0.2 | 0.1 | -5 | -0.1 | 0.3 | 207 | 0 | 0.5 | 718 |
| -0.1 | 0.1 | 11 | 0 | 0.3 | 488 | 0.1 | 0.5 | 1176 |
| 0 | 0.1 | 124 | 0.1 | 0.3 | 958 | 0.2 | 0.5 | 1755 |
| 0.1 | 0.1 | 590 | 0.2 | 0.3 | 1612 | 0.3 | 0.5 | 2436 |
| 0.2 | 0.1 | 1387 | 0.3 | 0.3 | 2389 | -0.3 | 0.6 | 48 |
| 0.3 | 0.1 | 2315 | -0.3 | 0.4 | 26 | -0.2 | 0.6 | 190 |
| -0.3 | 0.2 | -1 | -0.2 | 0.4 | 119 | -0.1 | 0.6 | 432 |
| -0.2 | 0.2 | 18 | -0.1 | 0.4 | 307 | 0 | 0.6 | 777 |
| -0.1 | 0.2 | 96 | 0 | 0.4 | 622 | 0.1 | 0.6 | 1230 |
| 0 | 0.2 | 316 | 0.1 | 0.4 | 1087 | 0.2 | 0.6 | 1788 |
| 0.1 | 0.2 | 785 | 0.2 | 0.4 | 1697 | 0.3 | 0.6 | 2447 |

## REFERENCES

1. I. S. Berezin and N. P. Zhidkov, Computing Methods, Vol. 2, Addison-Wesley, Reading, Mass., 1965, p. 448.

2. M. J. M. Bernal and J. R. Whiteman, "Numerical treatment of biharmonic boundary value problems with re-entrant boundaries," The Comp. Jour., vol. 13, 1970, p. 87.

3. S. D. Conte and R. T. Dames, "On an alternating direction method for solving the plate problem with mixed boundary conditions," Jour. A.C.M., vol. 7, 1960, p. 264.

4. D. Greenspan, "A numerical approach to biharmonic problems," The Comp. Jour., vol. 10, 1967, p. 198.

5. ____________, "Lectures on the Numerical Solution of Linear, Singular and Nonlinear Differential Equations, Prentice-Hall, Englewood Cliffs, N. J., 1968.

6. ____________, "Numerical studies of prototype cavity flow problems," The Comp. Jour., vol. 12, 1969, p. 89.

7. L. V. Kantorovich and V. I. Krylov, Approximate Methods of Higher Analysis, Noordhoff, Groningen, 1958, p. 61.

8. O. L. Mangasarian, "Numerical solution of the first biharmonic problem by linear programming," Int. Jour. Eng. Sci., vol. 1, 1963, p. 231.

9. E. O. Omarov, "An approximate solution to a partial differential equation of the type $u_{xxxx} + au_{xxyy} + bu_{yyyy} = f(x,y)$ by the method of lines in a trapezoidal region," Izv. Akad. Nauk Uz SSSR Ser. Fiz.-Mat. Nauk, No. 4, 1963, p. 33.

10. S. V. Parter, "On 'two-line' iterative methods for the Laplace and biharmonic difference equations," Num. Mat., vol. 1, 1959, p. 240.

11. E. M. Prikhodko, "Application of the method of summary representations to the numerical solution of biharmonic problems," Dopovidi Akad. Nauk Ukr. R.S.R., No. 7, 1964, p. 856.

12. J. Smith, "The coupled equation approach to the numerical solution of the biharmonic equation by finite differences," SIAM Jour. Num. Anal., 5, 1968, p. 323.

13. G. J. Tee, "A novel finite-difference approximation to the biharmonic operator," The Comp. Jour., vol. 6, 1963, p. 177.

14. M. L. Williams, "On the stress distribution at the base of a stationary crack," Jour. Appl. Mech., 24, 1957, p. 109.

15. E. Windsor, "Iterative solutions of biharmonic differential equations," Master's Thesis, New York University, 1957.

## APPENDIX - RE-ENTRANT BOUNDARY FORTRAN PROGRAM

```
      COMMON N,NPLUS1,M,MPLUS1
      DIMENSION PSI(70,70),OMA(70,70),SVPSI(70,70),SVOMA(70,70),
     1 SVOUT(70,70),VSS(70,70)
      RW=1
      F1=.2
      C1=.85
      REWIND 10
      READ 300,N,M,H
300   FORMAT(2I2,F8.2)
      MPLUS1=M+1
      MMESH=M-1
      NPLUS1=N+1
      NMESH=N-1
      MID=N/2+1
      MID1=MID+1
      MIDM=MID-1
      H2=H*H
      ISTOP=0
      EPS=.2
      EPSS=50.
C     INITIALIZE VECTORS
      NZ=0
      MIN=0
      MP=19
      R=0
104   CONTINUE
      JM=0
      KJ=0
      PRINT 2323,C1,F1
2323  FORMAT(1H1,2F8.3)
      ISTART =1
      ISTART=0
233   FORMAT(7F10.6)
      IF(ISTART .EQ. 1) GO TO 911
      NE=0
      NN=7
      DO 330 JP=1,NPLUS1,7
      LB=JP
      NE=NE+NN
      IF(NE .GT. NPLUS1) GO TO 331
      GO TO 332
331   NE=NPLUS1
332   DO 330 J=1,MPLUS1
      L=MPLUS1-J+1
      READ(10,233)(PSI(I,L),I=LB,NE)
330   CONTINUE
      NE=0
      NN=7
      DO 240 JP=1,NPLUS1,7
      LB=JP
      NE=NE+NN
      IF(NE .GT. NPLUS1) GO TO 241
      GO TO 242
241   NE=NPLUS1
242   DO 240 J=1,MPLUS1
      L=MPLUS1-J+1
```

```
      READ(10,233)(OMA(I,L),I=LB,NE)
240   CONTINUE
      PRINT 894,(OMA(I,3),I=31,39)
      PRINT 894,(OMA(I,2),I=31,39)
894   FORMAT(1X,9F11.2)
      PRINT 91
911   CONTINUE
      DO 2014 I=MID,N
2014   OMA(I,1)=OMA(I,3)
      CALL PRNTLT(PSI)
      CALL PRNTLT(OMA)
      REWIND 10
      DO 1001 I=2,N
      X=(I-1)*H
      X2=X*X
1001  PSI(I,MPLUS1)=10000.*X2/2.
      DO 1002 J=1,MPLUS1
1002  PSI(NPLUS1,J)=10000.*.8*.8/2.
      NM1=0
      NM2=0
      NM=0
      C2=1-C1
      F2=1-F1
C     BEGIN LOOP FOR OUTER ITERATIONS
23    DO 40 I=1,NPLUS1
      DO 40 J=1,MPLUS1
      VSS(I,J)=PSI(I,J)
40    SVOUT(I,J)=OMA(I,J)
      NM=NM+1
      NCOUNT=0
C   BEGIN INNER ITERATION FOR STREAM FUNCTION
11    DO 2 I=2,N
      DO 2 J=2,M
      IF(I .LE.MID.AND. J .EQ. 2) GO TO 2
      SVPSI(I,J)=PSI(I,J)
      IF(I .EQ. MIDI .AND. J .EQ. 3) GO TO 1428
      IF(I .EQ. MIDM .AND. J .EQ. 3) GO TO 1428
      IF(I .EQ. MID1 .AND. J .EQ. 2) GO TO 1428
      IF(I .EQ. MID .AND. J .EQ. 3) GO TO 1428
      PSI(I,J)=(-.8*PSI(I,J))+.45*(PSI(I,J-1)+PSI(I,J+1)+PSI(I-1,J)+
     1PSI(I+1,J)+H2*OMA(I,J))
      GO TO 2
1428  PSI(I,J)=(-.8*PSI(I,J )+.09*(4*PSI(I+1,J)+4*PSI(I,J+1)+4*PSI(I-1,J
     1)+4*PSI(I,J-1)+PSI(I+ ,J+1)+PSI(I-1,J+1)+PSI(I-1,J-1)+PSI(I+1,J-1)
     2+H2*OMA(I,J))
2     CONTINUE
      NM1=NM1+1
      DO 1003 I=MID,N
1003  PSI(I,1)=PSI(I,3)
C    TEST STREAM FUNCTION FOR CONVERGENCE
     DO 5 I=2,N
      DO 5 J=2,M
      DIFF=ABS(SVPSI(I,J)-PSI(I,J))
      IF(DIFF .GT.EPSS) GO TO 6
5     CONTINUE
```

```
C       RECALCULATE STREAM  FUNCTION USING WEIGHTING
        DO 114 I=2,N
        DO 114 J=2,M
114     PSI(I,J)=F1*VSS(I,J)+F2*PSI(I,J)
        DO 1114 I=MID,N
1114    PSI(I,1)=F1*VSS(I,1)+F2*PSI(I,1)
        GO TO 200
6       NCOUNT=NCOUNT+1
        IF(NCOUNT .GT. 100) GO TO 8
        GO TO 11
C       TEST STREAM FUNCTION FOR DIVERGENCE
8       IF(DIFF .GT. 10) GO TO 28
        PRINT 93
93      FORMAT(1H1,11H PSI VALUES)
        CALL PRNTLT(PSI)
10      FORMAT(10F11.6)
        NCOUNT=0
        GO TO 11
28      PRINT 81
81      FORMAT(13H PSI DIVERGED)
        CALL PRNTLT(PSI)
        CALL PRNTLT(OMA)
        GO TO 699
C       BEGIN INNER ITERATION FOR VORTICITY
200     NCOUNT=0
30      HCONST=C2*(-2./H2)
        DO 12 I=1,MIDM
12      OMA(I,2)=C1*OMA(I,2)+HCONST*PSI(I,3)
        OMA(MID,2)=C1*OMA(MID,2)+HCONST*(PSI(MID,3)+.5*PSI(MID1,2))
        DO 13 J=2,MPLUS1
13      OMA(1,J)=C1*OMA(1,J)+HCONST*PSI(2,J)
        DO 1012 I=2,N
1012    OMA(I,MPLUS1)=C1*OMA(I,MPLUS1)+    -C2*(-(4./H2)*PSI(I,MPLUS1)+(
       1I./H2)*PSI(I+1,MPLUS1)+(1./H2)*PSI(I-1,MPLUS1)+(2./H2)*PSI(I,M))
        DO 1013 J=2,M
1013    OMA(NPLUS1,J)=C1*OMA(NPLUS1,J)+    -C2*(-(4./H2)*PSI(NPLUS1,J)
       1+(1./H2)*PSI(NPLUS1,J+1)+(2./H2)*PSI(N,J)+(1./H2)*PSI(NPLUS1,J-1)+
       2(2./H)*10000.*.8)
90      CONTINUE
        NM2=NM2+1
        DO 14 I=2,N
        DO 14 J=2,M
        IF(I .LE.MID.AND. J .EQ. 2) GO TO 14
        A1=PSI(I+1,J)-PSI(I-1,J)
        B1=PSI(I,J+1)-PSI(I,J-1)
        A=ABS(A1)
        B=ABS(B1)
        W0=4+(A+B)*(R/2)
        IF(A1 .GE. 0) GO TO 15
        GO TO 16
15      W2=1+(R/2)*A
        W4=1
        GO TO 17
16      W2=1
        W4=1+A*(R/2)
17      IF(B1.GE. 0) GO TO 18
```

```
      GO TO 19
18    W1=1
      W3=1+B*(R/2)
      GO TO 20
19    W1=1+B*(R/2)
      W3=1
20    IF(ISTOP .EQ. 1) GO TO 425
      SVOMA(I,J)=OMA(I,J)
      GO TO 427
425   WM=ABS(SVOMA(I,J)-OMA(I,J))
      IF(WM .LT. WMIN) GO TO 426
      GO TO 427
426   WMIN=WM
427   CONTINUE
      IF(ISTOP .EQ. 1)GO TO 305
      IF(I .EQ. MID .AND. J .EQ. 3) GO TO 1429
      IF(I .EQ. MID1 .AND. J .EQ. 2) GO TO 1429
      IF(I .EQ. MIDM .AND. J .EQ. 3) GO TO 1429
      IF(I .EQ. MID1 .AND. J .EQ. 3) GO TO 1429
     OMA(I,J)=((W1/W0)*OMA(I+1,J)+(W2/W0)*OMA(I,J+1)+(W3/W0)*OMA(I-1,J)
    1+(W4/W0)*OMA(I,J-1))*RW+(1-RW)*OMA(I,J)
      GO TO 14
1429  OMA(I,J)=(1-RW)*OMA(I,J)+(RW/20.)*(4*OMA(I+1,J)+4*OMA(I,J+1)
    1+4*OMA(I-1,J)+4*OMA(I,J-1)+OMA(I+1,J+1)+OMA(I-1,J+1)+OMA(I-1,J-1)
    2+OMA(I+1,J-1))
      GO TO 14
305   DIFF   =((W1/W0)*OMA(I+1,J)+(W2/W0)*OMA(I,J+1)+(W3/W0)*OMA(I-1,J)
    1+(W4/W0)*OMA(I,J-1))-OMA(I,J)
      DIF=ABS(DIFF)
      IF(DIF .GT. EPS1) GO TO 282
      GO TO 421
282   EPS1=DIF
421   IF(DIF .LT. EPS2) GO TO 420
      GO TO 14
420    EPS2=DIF
14    CONTINUE
      DO 1014 I=MID,N
1014  OMA(I,1)=OMA(I,3)
      IF (ISTOP .EQ. 1) GO TO 700
C    TEST   VORTICITY     FOR CONVERGENCE
      DO 21 I=2,N
     DO 21 J=2,M
      IF(I .LE.MID.AND. J .EQ. 2) GO TO 21
      DIFF=ABS (SVOMA(I,J)-OMA(I,J))
      IF(DIFF .GE. EPS) GO TO 22
21   CONTINUE
C     RECALCULATE VORTICITY         USING WEIGHTING
      DO 414 I=2,N
      DO 414 J=2,M
414   OMA(I,J)=C1*SVOUT(I,J)+C2*OMA(I,J)
      DO 1414 I=MID,N
1414  OMA(I,1)=C1*SVOUT(I,1)+C2*OMA(I,1)
      PRINT 1016,NM,NM1,NM2
1016  FORMAT(1X,3I6)
      JM=JM+1
```

```
      IF(JM .EQ.30) GO TO 89
      GO TO 59
89    JM=0
      MIN=MIN+20
      PRINT 79,NM
79    FORMAT(1H1,I3,17H OUTER ITERATIONS)
      PRINT 91
      CALL PRNTLT(PSI)
      PRINT 92
      CALL PRNTLT(OMA)
      REWIND 10
      NZ=0
C     TEST OUTER ITERATIONS FOR CONVERGENCE
59    CONTINUE
      DO 45 I=2,N
      DO 45 J=2,M
      DIFF=ABS (SVOUT(I,J)-OMA(I,J))
      IF(DIFF .GT. EPS) GO TO 7
45     CONTINUE
       PRINT 99,NM
99    FORMAT(1H1,22H PROBLEM CONVERGED IN ,I4)
      PRINT 91
91    FORMAT(1X,11H PSI VALUES)
      CALL PRNTLT(PSI)
      PRINT 92
92    FORMAT(1H1,14H OMEGA  VALUES)
      CALL PRNTLT(OMA)
      EPS1=0
      RMAX=0
      EPS2=1
      WMIN=1
      RMIN=1
      ISTOP=1
      DO 181 II=2,N
      DO 181 JJ=2,M
      IF(I .LE.MID.AND. J .EQ. 2) GO TO 181
      RES=ABS (PSI(II,JJ)-VSS(II,JJ))
      IF(RES .GT. RMAX) GO TO 301
      GO TO 302
301   RMAX=RES
302   CONTINUE
      IF(RES .LT. RMIN) GO TO 422
      GO TO 320
422   RMIN=RES
320   CONTINUE
      A=-4*PSI(II,JJ)+PSI(II+1,JJ)+PSI(II,JJ+1)+PSI(II-1,JJ)+PSI(II,JJ-
     11)
      B=-H*H*OMA(II,JJ)
      D=ABS(A-B)
      IF(D .GT. EPS1) GO TO 182
      GO TO 183
182   EPS1=D
183   IF(D .LT. EPS2) GO TO 184
      GO TO 181
184   EPS2=D
181   CONTINUE
```

```
      PMAX=EPS1
      EPS1=0
      PMIN=EPS2
       EPS2=1
      GO TO 90
C     TEST OUTER ITERATIONS FOR DIVERGENCE
7     IF(DIFF .GT.9900) GO TO 199
      GO TO 23
22    NCOUNT=NCOUNT+1
      IF(NCOUNT .GT. 300) GO TO 24
      GO TO 90
C     TEST   VORTICITY      FOR DIVERGENCE
24    IF(DIFF .GT.999) GO TO 29
      PRINT 94
94    FORMAT(1H1,14H OMAEGA VALUES)
      CALL PRNTLT(OMA)
      PRINT 91
      CALL PRNTLT(PSI)
32    FORMAT(10F11.6)
      NCOUNT=0
      GO TO 90
29    PRINT 82
82     FORMAT(13H OMA DIVERGED)
      CALL PRNTLT(PSI)
      CALL PRNTLT(OMA)
      GO TO 699
199   PRINT 189
189   FORMAT(26H OUTER ITERATIONS DIVERGED)
700   CONTINUE
      PRINT 500,RMAX,RMIN
      PRINT 501,EPS,WMIN
500   FORMAT(2X,8H PSIMAX=,E12.4,8H PSIMIN=,E12.4)
501   FORMAT(2X,8H OMAMAX=,E12.4,8H OMAMIN=,E12.4)
      PRINT 502,PMAX,PMIN
502   FORMAT(2X,8HRPSIMAX=,E12.4,8HRPSIMIN=,E12.4)
      PRINT 503,EPS2,EPS1
503   FORMAT(2X,8HROMAMAX=,E12.4,8HROMAMIN=,E12.4)
699   CONTINUE
      SUBROUTINE PRNTLT(Z)
      COMMON N,NPLUS1,M,MPLUS1
      DIMENSION Z(70,70)
      IF(MIN .GT. MP) GO TO 5
      GO TO 6
5     NZ=NZ+2
      IF(NZ .GT. 6) GO TO 6
      NE=0
      NN=7
      DO 81 JP=1,NPLUS1,7
      LB=JP
      NE=NE+NN
      IF(NE .GT. NPLUS1) GO TO 10
      GO TO 12
10    NE=NPLUS1
12    DO 81 J=1,MPLUS1
      L=MPLUS1-J+1
```

```
81      WRITE(10,82)(Z(I,L),I=LB,NE)
82      FORMAT(7F10.1)
6       CONTINUE
        IF(N .GT. 11) GO TO 103
75      DO 61 J=1,MPLUS1
        L=MPLUS1-J+1
61      PRINT 52,(Z(I,L),I=1,NPLUS1)
       RETURN
103     NE=0
        NN=11
        DO 51 IP=1,NPLUS1,11
        IF( IP .GT. 1) GO TO 1
        GO TO 2
1       PRINT 3
3       FORMAT(////)
2       NB=IP
        NE=NE+NN
        IF(NE .GT. NPLUS1) GO TO 101
        GO TO 102
101      NE=NPLUS1
102     DO 51 J=1,MPLUS1
       L=MPLUS1-J+1
51      PRINT 52,(Z(I,L),I=NB,NE)
52      FORMAT(1X,11F10.3)
        RETURN
       END
```