Complex Polynomial Approximation of Two-Dimensional Potential Problems using Generalized Fourier Series

C.C. Yen and T.V. Hromadka II

Williamson and Schmid, Irvine, California 92714, USA

ABSTRACT

The objective of this paper is to present a new vector space norm which defines the best approximation of the two-dimensional Laplace equation in the mean square sense. The complex variable polynomial is now expanded as a generalized Fourier series. Because orthogonal functions are used, matrix solutions are eliminated; thus, considerably reducing the computational effort and memory requirements. Boundary conditions are approximated in a "mean-square" error sense in that a new vector space norm is defined which is analogous to the L_2 norm, and then minimized by the selection of complex coefficients to be associated to each term of the complex polynomial.

INTRODUCTION

The objective of this paper is to develop a new vector space norm (Hromadka, 1986) which is analogous to the L_2 norm. The numerical approach is to determine a complex variable polynomial which is now expanded as a generalized Fourier series--eliminating the matrix solution entirely. Boundary conditions are approximated in a "mean-square" error sense *in that a new vector space norm is defined* which is analogous to the L_2 norm, and then minimized by the selection of complex coefficients to be associated to each term of the complex polynomial. For engineering problems where the boundary condition values and their first derivative are piecewise continuous on Γ (i.e., Dirichlet conditions), the new complex polynomial approximation converges almost everywhere (ae) (Hromadka, 1986) on Γ as guaranteed by the well-known generalized Fourier series theory.

MODEL DEVELOPMENT

Given a complex function $\omega(z)$ analytic inside a simple closed contour Γ, Cauchy's theorem relates the value of $\omega(z_0)$ to the boundary integral (where Γ is a simple-closed contour enclosing a simply connected domain, Ω)

$$\omega(z_0) = \frac{1}{2\pi i} \oint_\Gamma \frac{\omega(z)}{(z - z_0)}\, dz, \quad i = \sqrt{-1} \tag{1}$$

where point (z_0) is interior of contour Γ, and contour integration is taken in the positive (counter-clockwise) direction.

The analytic function $\omega(z)$ is composed of two harmonic, real-variable, two-dimensional functions $\phi(x,y)$ and $\psi(x,y)$ where

$$\omega(z) = \phi(x,y) + i\,\psi(x,y) \tag{2}$$

in which $z = x + iy$, and $\phi(x,y)$ and $\psi(x,y)$ satisfy the well known Cauchy-Riemann relations (Mathews, 1982)

$$\frac{\partial\phi}{\partial x} = \frac{\partial\psi}{\partial y} \tag{3}$$

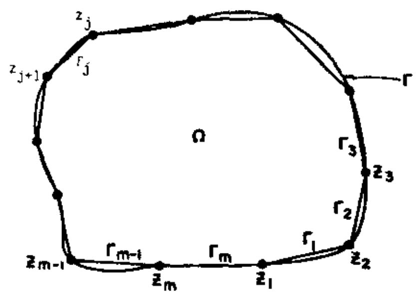
and

$$\frac{\partial\phi}{\partial y} = -\frac{\partial\psi}{\partial x} \tag{4}$$

An approximation of the Cauchy integral of (1) is given by

$$\oint_\Gamma \frac{\omega(z)}{(z - z_0)}\, dz \approx \oint_\Gamma \frac{\hat{\omega}(z)dz}{(z - z_0)} = \sum_j \int_{\Gamma_j} \frac{\hat{\omega}(z)dz}{(z - z_0)} \tag{5}$$

where each Γ_j is a segment of Γ (boundary segment) between two successive nodes on Γ (see Fig. 1) and z_0 *is located* interior of Ω (or arbitrarily close to Γ). In (5), the approximator $\hat{\omega}(z)$ can be extended as a complex polynomial of order (m-1) where 2m nodal points are specified on Γ. In this case, it is assumed that 2m boundary conditions are known along Γ, such as either ψ- or φ-values specified at each nodal point.



LEGEND

z_i: Nodal Coordinate for Node 1 ($z_{m+1} = z_1$)

Γ_i: Boundary Element Linking Nodes 1 and 2

Figure 1. Domain Ω ∪ Γ

The expansion of an order m complex polynomial is made for 2(m+1) nodal points by successive evaluation of

$$\hat{\omega}(z_j) = C_0 + C_1 z_j + C_2 z_j^2 + \cdots C_m z_j^m \tag{6}$$

0266-9463/86/020075 - 07 $2.00

where the C_j's are complex constants $C_j = \alpha_j + i\,\beta_j$; z_j are nodal points ($j = 1,2,\cdots,m$) defined on the problem boundary Γ (a simple closed contour). From (6) it is readily noted that $\hat{\omega}(z)$ is analytic over the entire complex plane.

The C_j of (6) are calculated in the L_2 norm sense by finding the best choice of C_j to minimize the mean-square error in matching the boundary condition values continuously along Γ. Notation is used for the known and unknown function values along Γ,

$$\left.\begin{aligned} \omega(\zeta) &= \Delta\xi_k(\zeta) + \Delta\xi_u(\zeta) \\ \hat{\omega}(\zeta) &= \Delta\hat{\xi}_k(\zeta) + \Delta\hat{\xi}_u(\zeta) \end{aligned}\right\} \zeta \in \Gamma \qquad (7)$$

where $\omega(z)$ is the solution to the boundary value problem over $\Omega \cup \Gamma$; $\hat{\omega}(z)$ is the complex polynomial approximation over $\Omega \cup \Gamma$; Δ is a descriptor function such that $\Delta = 1$ or i depending whether the associated $\xi_{k,u}$ or $\hat{\xi}_{k,u}$ function values are the real or imaginary term; and ζ is notation for the case of $z \in \Gamma$. Therefore, the objective is to compute the C_j which, for a given nodal distribution on Γ, minimize

$$I = ||\xi_k - \hat{\xi}_k||_2^2 = \int_\Gamma (\xi_k - \hat{\xi}_k)^2\, d\Gamma \qquad (8)$$

ORTHOGONAL COMPLEX POLYNOMIAL FUNCTIONS AND THE BEST APPROXIMATION

The complex polynomial approximation function of (6) can be written as

$$\hat{\omega}(z) = \sum_{j=1} c_j f_j \qquad (9)$$

where $f_j = (z_j + z_j^2 + \cdots + z_j^m)$. The Gram-Schmidt procedure can be used to orthogonalize the f_j such that

$$\hat{\omega}(z) = \sum_{j=1}^{m} \gamma_j g_j \qquad (10)$$

where γ_j are complex constants and

$$(g_j, g_k) = \int_\Gamma g_j g_k\, d\Gamma = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases} \qquad (11)$$

In (11), (g_j, g_k) is notation for the inner-product.

The boundary conditions on Γ are given by ξ_k where $\phi(\zeta)$ is known continuously on contour Γ_ϕ and $\psi(\zeta)$ is known continuously on Γ_ψ where $\Gamma_\phi + \Gamma_\psi = \Gamma$ and $\Gamma_\phi \cap \Gamma_\psi$ only on nodal points. The Γ_ϕ and Γ_ψ can be composed of a finite number of contours. Then the γ_j are computed which minimize

$$I = \int_{\Gamma_\phi} (\phi(\zeta) - \mathrm{Re}\,\Sigma\gamma_j g_j)^2 d\Gamma + \int_{\Gamma_\psi} (\psi(\zeta) - \mathrm{Im}\,\Sigma\gamma_j g_j)^2 d\Gamma \qquad (12)$$

Because the g_j are orthogonal, the γ_j are directly computed by

$$\gamma_j = (\xi_k, g_j)/(g_j, g_j) \qquad (13)$$

Then the best approximation (in the L_2 norm) is given by

$$\hat{\omega}(z) = \sum_{j=1}^{m} (\xi_k, g_j) g_j/(g_j, g_j) \qquad (14)$$

The c_j are then computed by back-substitution of the $\gamma_j g_j$ functions into the $c_j f_j$ functions. It is noted that by this approach, the c_j are computed directly without the use of a matrix system generation or matrix solution. This is important due to boundary integral methods (Brebbia, 1978; Liggett and Lui, 1983; Hromadka, 1984) resulting in the solution of fully populated, square matrix systems.

ORTHOGONAL VECTOR SYSTEMS AND THE BEST APPROXIMATION

Let F_j be linearly independent vectors of dimension n, for $j = 1,2,\cdots,m$. Then the Gram-Schmidt procedure can be used to construct orthongonal vectors G_j of dimension n such that the dot product gives

$$G_j \cdot G_k = \begin{cases} 1, & j = k \\ 0, & j \neq k \end{cases} \qquad (15)$$

Let B be a vector of dimension n. Then the best approximation of B in the subspace spanned by the G_j is given by the vector A where

$$A = \sum_{j=1}^{m} \eta_j G_j \qquad (16)$$

where

$$\eta_j = (B \cdot G_j)/(G_j \cdot G_j) \qquad (17)$$

The corresponding approximation to B with respect to the original F_j vectors is

$$A = \sum_{j=1}^{m} C_j F_j \qquad (18)$$

where the C_j are computed by back-substitution of $\eta_j G_j$ into the respective F_j components.

REPRESENTATION OF THE COMPLEX POLYNOMIAL APPROXIMATION FUNCTION BY A DIMENSION mn VECTOR SPACE

Let Γ be discretized into m boundary element Γ_j, $j = 1,2,\cdots,m$. On each element, define n interior evaluation points (usually evenly spaced), resulting in a total of mn points t_i on Γ. For each function f_j (see Eq. (9)), develop the vector F_j of dimension mn by

$$F_j = \{f_j(t_i);\ i = 1,2,\cdots,mn\} \qquad (19)$$

In (19), the coordinates of t_i are consistent for each vector F_j, $j = 1,2,\cdots,m$, such that points $(t_1, t_2, \cdots, t_n)$ occur in boundary element Γ_1. The resulting vectors F_j form the basis of a subspace F_{mn} where each vector $F \in F_{mn}$ is given by

$$F = \sum_{j=1}^{m} \eta_j F_j \qquad (20)$$

Similarly the boundary condition values defined on Γ can be represented by the vector B where

$$B = \{\xi_k(t_i);\ i = 1,2,\cdots,mn\} \qquad (21)$$

The best approximation of the vector B (in the ℓ_2 norm analogy of the L_2 norm) by a vector $A \in F_{mn}$ is given directly by (16) and (17). The corresponding estimate of the best approximation $\hat{\omega}(z)$ is given by

$$\hat{\omega}(z) = \sum_{j=1}^{m} \eta_j g_j \qquad (22)$$

Thus in the above, the best approximation for $\hat{\omega}(z)$ is estimated by using the best approximation from a vector space spanned by the vectors G_j. For more details and theory of this technique, refer to Hromadka (1986).

IMPLEMENTATION

A FORTRAN computer program was prepared which developed the best approximation in a vector space (of dimension mn) in order to estimate the c_j coefficients of Eq. (6). The basic steps used in the program are as follows:

1. Data entry of nodal point (m) coordinates and boundary values
2. Number of evaluation points entered (n)
3. Develop dimension mn vectors F_j, $j = 1,2,\cdots,m$
4. Develop dimension mn vector B of boundary values
5. Develop orthogonal vectors G_j, $j = 1,2,\cdots,m$
6. Compute vector coefficients η_j
7. Back substitute G_j vectors into F_j and compute the coefficients $\tilde{C}_j$; $j = 1,2,\cdots,\tilde{m}$
8. Define $c_j = \tilde{C}_j$ to determine the CVBEM approximation function, $\hat{\omega}(z)$.

It is noted that the $c_j = \alpha_j + i\beta_j$. Thus the above program steps involve two vectors for each C_j. That is from (1),

$$\hat{\omega}(z) = \sum_{j=1}^{m} \alpha_j[z_j + z_j^2 + \cdots + z_j^m] + \sum_{j=1}^{m} \beta_j[z_j + z_j^2 + \cdots + z_j^m] \qquad (23)$$

Hence the f_j vectors corresponding to the c_j have two separate components which are used, respectively, with the α_j and β_j. Consequently, for m nodes there are 2m coefficients to be computed.

COMPUTATIONAL EFFICIENCY

The use of the new CVBEM technique appears to reduce computational effort by approximately 20-percent in comparison to the referenced CVBEM model of Hromadka (1982). Current work is ongoing to evaluate the performance of this new technique in development of approximate boundaries (Hromadka, 1984) and incorporation into general engineering problems.

APPROXIMATE BOUNDARY DEVELOPMENT

Hromadka (1984) details the "approximate boundary" $\hat{\Gamma}$ technique for CVBEM error evaluation. The contour $\hat{\Gamma}$ represents the location where $\hat{\omega}(z)$ achieves the boundary conditions of $\omega(z)$ on Γ. That is, if the provided boundary conditions are level curves of $\omega(z)$ on Γ, then $\hat{\Gamma}$ represents the corresponding level curves of $\hat{\omega}(z)$. Hence if the approximate boundary $\hat{\Gamma}$ lies "sufficiently close" to Γ, the analyst can conclude that an adequate approximation has been developed. This error evaluation technique is very useful due to the ease of interpretation. Even beginners can develop highly accurate complex polynomial approximations by simply observing the relationship of $\hat{\Gamma}$ to Γ, and adding nodes to Γ where departures are considered unacceptable. In the included example problems, approximate boundaries are developed for each test problem.

APPLICATIONS

A FORTRAN computer program was prepared based on the least-square boundary fit described in the previous sections. Matrix solution routines are not needed due to the orthonormal vector technique.

By entering x,y-coordinates, $\hat{\omega}(z)$ values are computed and the flow-net can be plotted with respect to the approximation $\hat{\phi}(z)$ and $\hat{\psi}(z)$ values.

Example 1. Ideal fluid flow around a cylindrical corner has the analytic solution of $\omega(z) = z^2 + z^{-2}$. Figure 2 depicts the problem geometry and specified boundary conditions and shows the complex polynomial computed flow net.

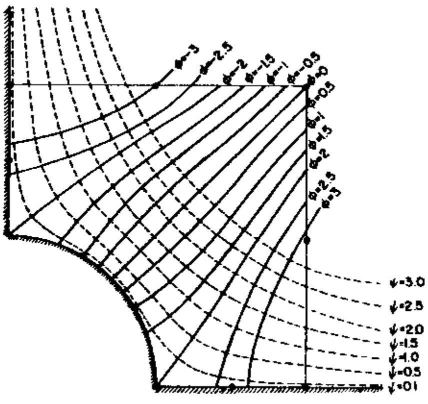


Figure 2. Computed Flow Net for Ideal Fluid Flow Around a Cylindrical Corner

Example 2. By taking the advantage of the symmetric geometry of an elliptical section (see Fig. 3), let's consider the St. Venant torsion problem for the quartered solid illiptical cross section (see Fig. 4). The analytic solution $\phi(x,y)$ and associated shear stresses are given by

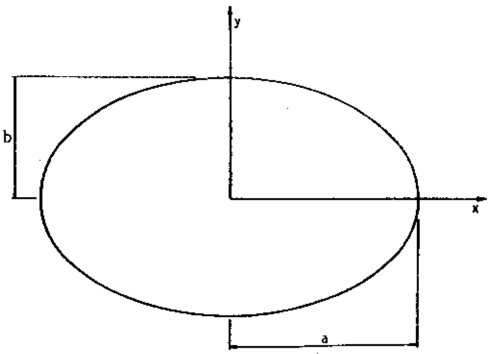


Figure 3. Elliptical Section Geometry

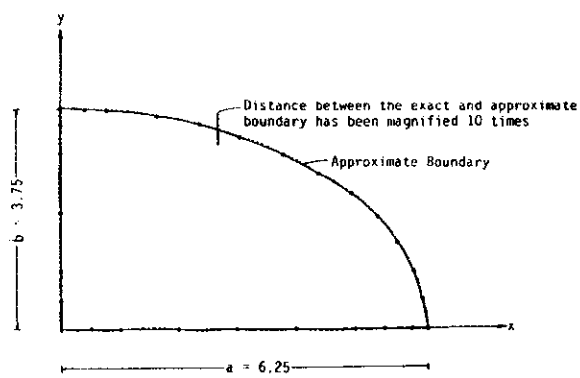


Figure 4. CVBEM Approximate Boundary for Quartered Elliptical Section

$$\phi(x,y) = (x^2 + y^2)/2 - a^2b^2(x^2/a^2 + y^2/b^2 - 1)/(a^2 + b^2)$$

$$\tau_{xz} = -G\theta(2ya^2)/(a^2 + b^2)$$

$$\tau_{yz} = G\theta(2xb)/(a^2 + b^2)$$

Figure 4 shows the complex polynomial approximate boundary for quartered elliptical section.

Example 3. Figure 5 shows streamlines and equipotential lines for soil-water flow through a homogeneous soil. *The locations of the phreatic surface and the seepage* face can be easily determined by the approximate boundary technique.

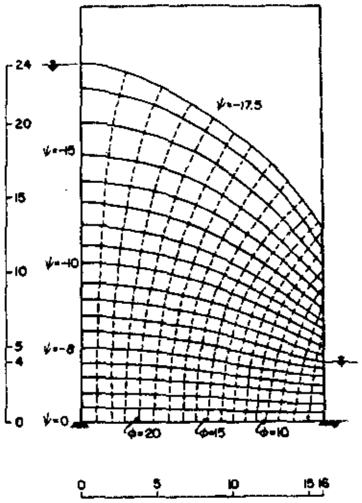


Figure 5. *Computed Flow Net for Soil-Water through a Homogeneous Soil*

REFERENCES

1. Brebbia, C. A., "The Boundary Element Method for *Engineers*," *Pentech Press*, 1978.

2. Hromadka II, T. V., "The Complex Variable Boundary *Element Method*," *Springer-Verlag*, 1984.

3. Hromadka II, T. V., "A Boundary Integration Boundary Method Without Matrices," *Present to* A.S.M.E. OMAE Conference, Tokyo, Japan, 1986.

4. Liggett, J. A. and Lui, P. L-F., "The Boundary Integral Equation Method for Porous Media Flow," London, UK, George Allen & Unwin, Ltd., 1983.

5. Mathews, J. H., "Basic Complex Variables for Mathematics and Engineering," Allyn and Bacon, Inc., 1982.

APPENDIX

Complex Polynomial Approximation Model

DATA PREPARATION

Data is entered into file LTWO.DAT and the program solutions are contained in file LTWO.ANS. Data entry is as follows:

Line #	Variables
1	NNOD, NTEST, KODE
2	X(1),Y(I),KTYPE(I),VALUE(1,1), VALUE(1,2)
.	
.	
.	
NNOD+1	X(NNOD),Y(NNOD),KTYPE(NNOD), VALUE(NNOD,1),VALUE(NNOD,2)

DESCRIPTION OF VARIABLES

NNOD = Total number of nodes

NTEST = Number of evaluation points between each nodal point

KODE = { 0 , Summary results; 1 , Detailed results

X(I) = x-coordinate for node I

Y(I) = *y-coordinate for node I*

KTYPE(I) = { 1 , state variable specified; 2 , stream function specified; 3 , both state variable and stream .function are known

VALUE(I,1) = specified state variable *value for node I*

VALUE(I,2) = specified stream function value for node I

PROGRAM LISTING

```
C
C  MAIN PROGRAM
C
C  THIS IS A GENERALIZE FOURIER SERIES ANALYSIS
C  WHICH USES THE POLYNOMIAL FUNCTION TO SOLVE THE
C  LAPLACE EQUATION
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/BLK 1/ X(80),Y(80)
      COMMON/BLK 2/ ANGLE(80),KTYPE(80)
      COMMON/BLK 3/ VALUE(80,2)
      COMMON/BLK 4/ WB(60),B(60)
      COMMON/BLK 5/ G(45,60)
      COMMON/BLK 6/ XX(45,45)
C
C  OPEN DATA FILES
C
      NRD=1
      NWT=2
      NWD=3
      NTMR=11
      NTMW=10
600   OPEN NRD,'LTWO.DAT'
      OPEN NWT,'LTWO.ANS'
C
C READ INPUT DATA
C
      PI=3.141592653
      READ FREE(NRD) NNOD,NTEST,KODE
610   DO 7 I=1,NNOD
      READ FREE(NRD) X(I),Y(I),KTYPE(I),VALUE(1,1),VALUE(1,2)
7     CONTINUE
      CALL ANG(NNOD)
615   WRITE(NWT,2)
      DO 9 I=1,NNOD
      IF(KTYPE(I).EQ.1)VALUE(I,2)=0.
      IF(KTYPE(I).EQ.2)VALUE(I,1)=0.
      WRITE(NWT,8) I,X(I),Y(I),KTYPE(I),VALUE(1,1),VALUE(1,2)
     1             ,ANGLE(I)
9     CONTINUE
C
C  OUTPUT FORMATS
C
2     FORMAT(//,10X,'*** BOUNDARY NODE DATA ***',
     1//,1X,'NODE',3X,'X(I)',6X,'Y(I)',4X,'KTYPE(I)',5X,
     2'VAUE(I)',9X,/,2X,'NO.',21X,'1=SV;2=SF',3X,
     3'SV',5X,'SF',8X,/,27X,'3=SV&SF',20X,'ANGLE(I)')
3     FORMAT(2X,2I3,3X,D10.4)
4     FORMAT(/,10X,'*** APPROXIMATE NODAL VALUES AND ERRORS ***',
     1//,6X,'NODE',9X,'STATE',12X,'STREAM',/,
     25X,'NUMBER',6X,'VARIABLE',10X,'FUNCTION',12X,'ERROR',/)
5     FORMAT(3X,I5,3(8X,D10.4))
8     FORMAT(1X,I3,2X,F8.3,2X,F8.3,5X,I2,4X,F7.2,1X,F7.2,5X,F6.2)
21    FORMAT(//,10X,'*** EVALUATION POINT DATA ***',
     1//,1X,'POINT',2X,'X(I)',6X,'Y(I)',4X,'KTYPE(I)',5X,
     2'VALUE(I)',/,2X,'NO.',21X,'1=SV;2=SF',3X,'SV',5X,'SF',/,
     327X,'3=SV&SF')
```

```
22    FORMAT(10(1X,D10.4))
23    FORMAT(/,10X,'*** NODE NUMBER #',I3,2X,'(REAL,IMAGINARY)')
24    FORMAT(/,10X,'*** NODAL POINT VECTOR EXPANSION, F(I) ***',/)
25    FORMAT(/,10X,'*** ORTHOGONAL VECTOR EXPANSION, G(I) ***',/)
26    FORMAT(/,10X,'*** ORTHOGONAL TEST (G(I),G(J)) ***',//,
     1'    I  J  (G(I),G(J))')
28    FORMAT(/,10X,'*** EVALUATION COEFFICIENTS,',
     1' (G(I),B(I))/(G(I),G(I)) ***',/)
29    FORMAT(/,10X,'*** BACKSUBSTITUTION COEFFICIENTS ***',/)
31    FORMAT(/,2X,'ENTER A [1] FOR ADDING AN ADDITIONAL NODAL POINT')
32    FORMAT(/,2X,'ENTER THE NODE NUMBER THAT',/,
     12X,'THE ADDITIONAL NODAL POINT WILL FOLLOWED')
33    FORMAT(/,2X,'ENTER THE X- AND Y- COORDIATES',/,
     12X,'FOR THE ADDITIONAL NODAL POINT')
34    FORMAT(/,2X,'THE ADDITIONAL NODAL POINT (',F8.4,',',F8.4,
     1') ',/,'  HAS THE NORM EQUAL TO ',D10.4)
35    FORMAT(/,2X,'ENTER A [1] TO ACCEPT THIS ADDITIONAL NODAL POINT')
36    FORMAT(/,2X,'*** BESSEL INEQUALITY ***',/,2X,D10.4,
     1' >= ',D10.4,' AND THE DIFFERENCE IS ',D10.4,/,72('-'),/)
38    FORMAT(2(1X,F10.4),1X,I2,3(1X,F10.4))
45    FORMAT(/,10X,'*** APPROXIMATE POINT VALUES AND ERRORS --',
     1' FUNCTION FORM ***',//,5X,'POINT',9X,'STATE',12X,'STREAM',/,
     25X,'NUMBER',6X,'VARIABLE',10X,'FUNCTION',12X,'ERROR',/)
71    FORMAT(2X,'ENTER A [1] TO EVALUATE THE STATE VARIABLE AND',
     1/,'  STREAM FUNCTION FOR A GIVEN POINT')
72    FORMAT(2X,'===> EXECUTE PROGRAM "LTWOPI" TO EVALUATE',/,
     1'  STATE VARIABLE AND STREAM FUNCTION FOR A GIVEN POINT')
81    FORMAT(1X,I3,2X,F8.3,2X,F8.3,5X,I2,4X,F7.2,1X,F7.2)
99    FORMAT(/,72('='))
C
C  CALCULATE X-COORINATE, Y-COORIDNATE, AND BOUNDARY VALUES
C  FOR EVALUATION POINTS
C
650   K=NNOD
307   DO 300 I=1,NNOD
      IP1=I+1
      IF(IP1.GT.NNOD)IP1=1
      X1=X(I)
      Y1=Y(I)
      X2=X(IP1)
      Y2=Y(IP1)
      DX=X2-X1
      DY=Y2-Y1
      R11=VALUE(I,1)
      R12=VALUE(I,2)
      R21=VALUE(IP1,1)
      R22=VALUE(IP1,2)
      DR1=R21-R11
      DR2=R22-R12

      DO 300 J=1,NTEST
      K=K+1
      P=1./(2.*(NTEST+1))
      IF(NTEST.EQ.1) RATIO=0.5
      IF(NTEST.GT.1) RATIO=P+(1.-2.*P)*FLOAT(J-1)/(FLOAT(NTEST)-1.)
      X(K)=X1+DX*RATIO
      Y(K)=Y1+DY*RATIO
      VALUE(K,1)=R11+DR1*RATIO
      VALUE(K,2)=R12+DR2*RATIO
      KTYPE(K)=MIN0(KTYPE(I),KTYPE(IP1))
300   CONTINUE
      NTOT=NNOD*NTEST
      NTOTL=2*NNOD
617   WRITE(NWT,21)
      SAREA=0.
      DO 310 KK=1,NTOT
      I=KK+NNOD
      IF(KTYPE(I).EQ.1)VALUE(I,2)=0.
      IF(KTYPE(I).EQ.2)VALUE(I,1)=0.
      SAREA=SAREA+VALUE(I,1)**2+VALUE(I,2)**2
      WRITE(NWT,81)KK,X(I),Y(I),KTYPE(I),VALUE(I,1),VALUE(I,2)
      ANGLE(I)=ANGLE(I)*PI/180.
310   CONTINUE
C
C  DETERMINE THE COEFFICIENTS OF ALPHA'S AND BETA'S FOR
C  COMPLEX VARIABLE POLYNOMIAL Z**N ARITEMATIC
C
      IF(KODE.EQ.1)WRITE(NWT,99)
      IF(KODE.EQ.1)WRITE(NWT,24)
      DO 320 I=1,NNOD
      II=2*I
      IIM1=II-1
      XX=X(I)
      YY=Y(I)
      DO 330 KK=1,NTOT
      J=KK+NNOD
      XJ=X(J)
      YJ=Y(J)
      A1=XJ-XX
      B1=YJ-YY
      RJ=DSQRT(A1*A1+B1*B1)
      CALL CAUCH5(A1,B1,TH)
      TH=FLOAT(I)*TH
      ALPHA=RJ**I*DCOS(TH)
      BETA=RJ**I*DSIN(TH)
337 * GO TO(335,345,335)KTYPE(J)
335   G(IIM1,KK)=ALPHA
      G(II,KK)=-1.*BETA
     *GO TO 330
345   G(IIM1,KK)=BETA
      G(II,KK)=ALPHA
330   CONTINUE
      IF(KODE.NE.1)GO TO 320
      WRITE(NWT,23)I
      WRITE(NWT,22)(G(IIM1,K),K=1,NTOT)
      WRITE(NWT,22)(G(II,K),K=1,NTOT)
320   CONTINUE
C
C  USE THE GRAM-SCHMIDT ORTHONORANLIZATION PROCESS
C  TO DETERMINE SERIES OF ORTHOGONAL VECTORS
C
      DO 20 I=2,NTOTL
      DO 20 KK=2,I
      SUM1=0.
      SUM2=0.
      DO 30 J=1,NTOT
      SUM1=SUM1+G(I,J)*G(KK-1,J)
      SUM2=SUM2+G(KK-1,J)*G(KK-1,J)
30    CONTINUE
      XKK=-1.*SUM1/SUM2
      XK(I,KK-1)=XKK
      DO 40 J=1,NTOT
      G(I,J)=G(I,J)+XKK*G(KK-1,J)
40    CONTINUE
20    CONTINUE
      IF(KODE.NE.1)GO TO 55
      WRITE(NWT,25)
      DO 47 I=1,NTOTL
47    WRITE(NWT,22)(G(I,J),J=1,NTOT)
C..CHECK ORTHOGONALITY OF VECTORS G(I)
      WRITE(NWT,26)
55    DO 50 I=1,NTOTL
      IP1=I+1
      IF(I.EQ.NTOTL)GO TO 80
      DO 70 K=IP1,NTOTL
      SUM=0.
      DO 60 J=1,NTOT
60    SUM=SUM+G(I,J)*G(K,J)
      IF(KODE.EQ.1)WRITE(NWT,3)I,K,SUM
      IF(KODE.NE.1 .AND. ABS(SUM).GT..00001)WRITE(NWT,3)I,K,SUM
70    CONTINUE
50    CONTINUE
80    CONTINUE
C..COMPUTE THE COEFFICIENTS OF B(I)=(W,G(I))/(G(I),G(I))
      IF(KODE.EQ.1)WRITE(NWT,28)
      SUM=0.
      DO 120 I=1,NTOTL
      BK1=0.
      BK2=0.
      DO 130 KK=1,NTOT
      J=KK+NNOD
      IF(KTYPE(J).EQ.1 .OR. KTYPE(J).EQ.3)BK1=BK1+VALUE(J,1)*G(I,KK)
      IF(KTYPE(J).EQ.2)BK1=BK1+VALUE(J,2)*G(I,KK)
      BK2=BK2+G(I,KK)*G(I,KK)
130   CONTINUE
C..COMPUTE THE NORM OF THE GENERALIZED FOURIER COEFFICIENTS
      B(I)=BK1/BK2
      SUM=SUM+BK1*BK1/BK2
120   CONTINUE
      IF(KODE.EQ.1)WRITE(NWT,22)(B(I),I=1,NTOTL)
C..COMPUTE THE COEFFICIENTS OF THE APPROXIMATE FUNCTIONS
660   DO 200 I=NTOTL,1,-1
      IF(I.EQ.NTOTL)XK(NTOTL,I)=B(NTOTL)
      IF(I.NE.NTOTL)XK(NTOTL,I)=XK(NTOTL,I)*B(NTOTL)+B(I)
200   CONTINUE

      NTOT1=NTOTL-1
      DO 210 I=NTOT1,1,-1
      DO 210 J=I,1,-1
      IF(I.EQ.J)GO TO 210
      IF(I.NE.J)XK(NTOTL,J)=XK(NTOTL,I)*XK(I,J)+XK(NTOTL,J)
210   CONTINUE
      IF(KODE.EQ.1)WRITE(NWT,29)
      IF(KODE.EQ.1)WRITE(NWT,22)(XK(NTOTL,I),I=1,NTOTL)
C..APPROXIMATE THE EVALUATION POINT -- FUNCTION FORM
      WRITE(NWT,99)
      WRITE(NWT,45)
      DO 480 I=1,NTOT
      II=I+NNOD
      XX=X(II)
      YY=Y(II)
      XRE=0.
      XIM=0.
      DO 470 J=1,NNOD
      JJ=J*2
      JJM1=JJ-1
      XJ=X(J)
      YJ=Y(J)
      A1=XX-XJ
      B1=YY-YJ
      RJ=DSQRT(A1*A1+B1*B1)
      CALL CAUCH5(A1,B1,TH)
      TH=FLOAT(J)*TH
      ALPHA=RJ**J*DCOS(TH)
      BETA=RJ**J*DSIN(TH)
475   XRE=XRE+XK(NTOTL,JJM1)*ALPHA-XK(NTOTL,JJ)*BETA
      XIM=XIM+XK(NTOTL,JJM1)*BETA+XK(NTOTL,JJ)*ALPHA
470   CONTINUE
      XD=XRE-VALUE(II,1)
      IF(KTYPE(II).EQ.2)XD=XIM-VALUE(II,2)
      WRITE(NWT,5)I,XRE,XIM,XD
480   CONTINUE
C..APPROXIMATE THE NODAL VALUES -- FUNCTION FORM
290   WRITE(NWT,99)
      WRITE(NWT,4)
      DO 280 I=1,NNOD
      XX=X(I)
```

```
      YY=Y(I)
      XRE=0.
      XIM=0.
      DO 270 J=1,NNOD
      IF(I.EQ.J)GO TO 270
      JJ=J*2
      JJM1=JJ-1
      XJ=X(J)
      YJ=Y(J)
      A1=XX-XJ
      B1=YY-YJ
      RJ=DSQRT(A1*A1+B1*B1)
      CALL CAUCH5(A1,B1,TH)
      TH=FLOAT(J)*TH
      ALPHA=RJ**J*DCOS(TH)
      BETA=RJ**J*DSIN(TH)
275   XRE=XRE+XK(NTOTL,JJM1)*ALPHA-XK(NTOTL,JJ)*BETA
      XIM=XIM+XK(NTOTL,JJM1)*BETA+XK(NTOTL,JJ)*ALPHA
270   CONTINUE
      XD=XRE-VALUE(I,1)
      IF(KTYPE(I).EQ.2)XD=XIM-VALUE(I,2)
      WRITE(NWT,5)I,XRE,XIM,XD
280   CONTINUE
      WRITE(NWT,99)
C..BESSEL'S INEQUALITY
      DIFF=SAREA-SUM
      IF(ABS(DIFF).LT.0.00001)DIFF=0.
      WRITE(NWT,36)SAREA,SUM,DIFF
      WRITE(NTMW,71)
      READ FREE(NTMR) KEVA
      IF(KEVA.NE.1)GOTO 700
      OPEN NWD,"LTWO1.DAT"
      WRITE FREE(NWD) NNOD,NNOD,NTOTL
      DO 710 I=1,NNOD
710   WRITE FREE(NWD) X(I),Y(I),ANGLE(I)
      WRITE FREE(NWD) (XK(NTOTL,I),I=1,NTOTL)
      CLOSE NWD
      WRITE(NTMW,72)
700   CLOSE NRD
      CLOSE NWT
      STOP
      END
```

APPROXIMATE BOUNDARY MODEL PROGRAM

```
C
C  MAIN PROGRAM
C
C  THIS PROGRAM EVALUATES THE STATE VARIABLE AND STREAM FUNCTION
C  FOR A GIVEN NODAL POINT
C
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      COMMON/BLK 1/ X(50),Y(50)
      COMMON/BLK 2/ ANGLE(50),XK(50)
C
C  OPEN DATA FILES
C
      NRD=1
      NWD=2
      NTR=11
      NTW=10
      OPEN NRD,"LTWO1.DAT"
C
C READ INPUT DATA
C
      PI=3.141592653
      READ FREE(NRD) NNOD,NNODP,NTOTL
      DO 10 I=1,NNOD
      READ FREE(NRD) X(I),Y(I),ANGLE(I)
10    CONTINUE
      READ FREE(NRD) (XK(I),I=1,NTOTL)
15    CLOSE (NRD)
      DO 18 I=1,NNOD
      ANGLE(I)=ANGLE(I)*PI/180.
18    CONTINUE
C
C  OUTPUT FORMATS
C
4     FORMAT(/,2X,'*** INVALID DATA ENTRY...TRY AGAIN ***',/)
5     FORMAT(3X,F10.4,2X,F10.4,2X,2(2X,F10.3))
6     FORMAT(10X,'*** APPROXIMATE SOLUTION ***',/)
7     FORMAT(2X,'EVALUATE THE STATE VARIABLE AND STREAM FUNCTION',
     1' FOR A GIVEN POINT',/,2X,'ENTER THE X-COORDINATE')
8     FORMAT(2X,'ENTER THE Y-COORDINATE')
9     FORMAT(/,7X,'X-',10X,'Y-',10X,'STATE',6X,'STREAM',
     1/,3X,'COORDINATE',2X,'COORDINATE',4X,'VARIABLE',4X,
     2'FUNCTION',/)
11    FORMAT(/,2X,'ENTER THE OPTION :',/,3X,
     1'(1) FOR ANOTHER POINT',/,3X,'(2) FOR ACCEPTING THE'
     2,' CURRENT POINT AND STARTING A NEW POINT',/,3X,
     3'(3) FOR ACCEPTING THE CURRENT POINT AND TERMINATING'
     4,' THE PROCESS')
12    FORMAT(/,2X,'*** THE NODAL POINT HAS BEEN USED...TRY AGAIN'
     1,' ***',/)
C
C  APPROXIMATE THE STATE VARIABLE AND STREAM FUNCTION
C
      OPEN NWD,"LTWO1.ANS"
      WRITE(NWD,6)
      WRITE(NWD,9)
300   WRITE(NTW,7)
      READ FREE(NTR) XX
      WRITE(NTW,8)
      READ FREE(NTR) YY
      DO 100 I=1,NNOD
      IF(XX.EQ.X(I) .AND. YY.EQ.Y(I))GO TO 110
100   CONTINUE
      GO TO 120
110   WRITE(NTW,12)
      GO TO 300
120   XRE=0.
      XIM=0.
      DO 270 J=1,NNOD
      JJ=J*2
      JJM1=JJ-1
      A1=XX-X(J)
      B1=YY-Y(J)
      RJ=DSQRT(A1*A1+B1*B1)
      CALL CAUCH5(A1,B1,ANG)
      TH=FLOAT(J)*ANG
      ALPHA=RJ**J*DCOS(TH)
      BETA=RJ**J*DSIN(TH)
275   XRE=XRE+XK(JJM1)*ALPHA-XK(JJ)*BETA
      XIM=XIM+XK(JJM1)*BETA+XK(JJ)*ALPHA
270   CONTINUE
      WRITE(NTW,9)
      WRITE(NTW,5)XX,YY,XRE,XIM
330   WRITE(NTW,11)
      READ FREE(NTR) KODE
      IF(KODE.EQ.1) GO TO 300
      IF(KODE.EQ.2) GO TO 310
      IF(KODE.EQ.3) GO TO 320
      WRITE(NTW,4)
      GO TO 330
310   WRITE(NWD,5)XX,YY,XRE,XIM
      GO TO 300
320   WRITE(NWD,5)XX,YY,XRE,XIM
      CLOSE NWD
      STOP
      END
```

SUBROUTINES

```
      SUBROUTINE ANG(NNOD)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
      COMMON/BLK 1/ X(80),Y(80)
      COMMON/BLK 2/ ANGLE(80),KTYPE(80)
C
C     THIS SUBROUTINE CALCULATES THE ANGLE BETWEEN NODAL POINTS
C
      DO 100 I=1,NNOD
      J=I-1
      JJ=I+1
      IF(J.EQ.0)J=NNOD
      IF(JJ.GT.NNOD)JJ=1
      XJ=X(J)-X(I)
      YJ=Y(J)-Y(I)
      XJJ=X(JJ)-X(I)
      YJJ=Y(JJ)-Y(I)
      CALL CAUCH5(XJJ,YJJ,AJJ)
      CALL CAUCH5(XJ,YJ,AJ)
      ANGLE(I)=(AJ-AJJ)*180./3.141592653
      IF(ANGLE(I).LT.0.)ANGLE(I)=ANGLE(I)+360.
100   CONTINUE
      RETURN
      END

C------------------------------------------------------------------
C     SUBROUTINE CAUCH5
C------------------------------------------------------------------
      SUBROUTINE CAUCH5(X,Y,ANGLE)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
C
C  THIS SUBROUTINE DETERMINES THE POSITIVE ANGLE
C  OF COMPLEX POINT Z WITH RESPECT TO THE ORIGIN
C
      PI=3.141592653
      IF(X.EQ.0. .AND. Y.GT.0.)ANGLE=.5*PI
      IF(X.EQ.0. .AND. Y.LT.0.)ANGLE=1.5*PI
      IF(X.GT.0. .AND. Y.GE.0.)ANGLE=DATAN(Y/X)
      IF(X.LT.0. .AND. Y.GE.0.)ANGLE=PI-DATAN(-Y/X)
      IF(X.LT.0. .AND. Y.LT.0.)ANGLE=PI+DATAN(Y/X)
      IF(X.GT.0. .AND. Y.LT.0.)ANGLE=2.*PI-DATAN(-Y/X)
      IF(X.EQ.0. .AND. Y.EQ.0.)ANGLE=0.
      RETURN
      END
C
```

EXAMPLE INPUT FILE

```
11 3 0
 0.00 1.00 3  0 0
 .5 .866 2 0 0
 .866 .5 2 0 0
 1. 0 2 0 0
 1.5 0 2 0 0
 2 0 3 2.5 0
 2 1 1 2.4 0
 2 2 1 2.25 0
 1 2 1 1.2 0
 0 2 1 0 0
 0 1.5 1 0 0
```

EXAMPLE OUTPUT FILE

*** BOUNDARY NODE DATA ***

NODE NO.	X(I)	Y(I)	KTYPE(I) 1=SV;2=SF 3=SV&SF	VAUE(I) SV	SF	ANGLE(I)
1	.000	1.000	3	.00	.00	105.00
2	.500	.866	2	.00	.00	210.00
3	.866	.500	2	.00	.00	210.00
4	1.000	.000	2	.00	.00	105.00
5	1.500	.000	2	.00	.00	180.00
6	2.000	.000	3	2.50	.00	90.00
7	2.000	1.000	1	2.40	.00	180.00
8	2.000	2.000	1	2.25	.00	90.00
9	1.000	2.000	1	1.20	.00	180.00
10	.000	2.000	1	.00	.00	90.00
11	.000	1.500	1	.00	.00	180.00

*** EVALUATION POINT DATA ***

POINT NO.	X(I)	Y(I)	KTYPE(I) 1=SV;2=SF 3=SV&SF	VALUE(I) SV	SF
1	.063	.983	2	.00	.00
2	.250	.933	2	.00	.00
3	.438	.883	2	.00	.00
4	.546	.820	2	.00	.00
5	.683	.683	2	.00	.00
6	.820	.546	2	.00	.00
7	.883	.438	2	.00	.00
8	.933	.250	2	.00	.00
9	.983	.063	2	.00	.00
10	1.063	.000	2	.00	.00
11	1.250	.000	2	.00	.00
12	1.438	.000	2	.00	.00
13	1.563	.000	2	.00	.00
14	1.750	.000	2	.00	.00
15	1.938	.000	2	.00	.00
16	2.000	.125	1	2.49	.00
17	2.000	.500	1	2.45	.00
18	2.000	.875	1	2.41	.00
19	2.000	1.125	1	2.38	.00
20	2.000	1.500	1	2.33	.00
21	2.000	1.875	1	2.27	.00
22	1.875	2.000	1	2.12	.00
23	1.500	2.000	1	1.73	.00
24	1.125	2.000	1	1.33	.00
25	.875	2.000	1	1.05	.00
26	.500	2.000	1	.60	.00
27	.125	2.000	1	.15	.00
28	.000	1.938	1	.00	.00
29	.000	1.750	1	.00	.00
30	.000	1.563	1	.00	.00
31	.000	1.438	1	.00	.00
32	.000	1.250	1	.00	.00
33	.000	1.063	1	.00	.00

*** APPROXIMATE POINT VALUES AND ERRORS -- FUNCTION FORM ***

POINT NUMBER	STATE VARIABLE	STREAM FUNCTION	ERROR
1	.1247D 00	.8132D-02	.8132D-02
2	.5133D 00	-.2645D-01	-.2645D-01
3	.8741D 00	.1551D-01	.1551D-01
4	.1087D 01	.1596D-01	.1596D-01
5	.1383D 01	-.1492D-01	-.1492D-01
6	.1629D 01	.5478D-02	.5478D-02
7	.1756D 01	.1635D-02	.1635D-02
8	.1899D 01	-.1030D-01	-.1030D-01
9	.1959D 01	.6524D-02	.6524D-02
10	.1962D 01	.5542D-03	.5542D-03
11	.2024D 01	-.5064D-02	-.5064D-02
12	.2113D 01	.2246D-02	.2246D-02
13	.2180D 01	.3378D-02	.3378D-02
14	.2303D 01	-.3830D-02	-.3830D-02
15	.2457D 01	.1139D-02	.1139D-02
16	.2487D 01	.1106D 00	-.3371D-03
17	.2450D 01	.3844D 00	.3415D-03
18	.2412D 01	.6929D 00	-.7094D-03
19	.2381D 01	.9189D 00	.6956D-04
20	.2325D 01	.1286D 01	.1261D-03
21	.2268D 01	.1660D 01	-.3985D-03
22	.2119D 01	.1781D 01	.5664D-03
23	.1724D 01	.1702D 01	-.1250D-02
24	.1334D 01	.1646D 01	.2518D-02
25	.1047D 01	.1612D 01	-.2520D-02
26	.6012D 00	.1550D 01	.1171D-02
27	.1497D 00	.1515D 01	-.3129D-03
28	-.5643D-03	.1432D 01	-.5643D-03
29	.2169D-02	.1200D 01	.2169D-02
30	-.1249D-02	.9489D 00	-.1249D-02
31	-.1560D-02	.7689D 00	-.1560D-02
32	.1468D-02	.4806D 00	.1468D-02
33	.4714D-03	.1596D 00	.4714D-03

*** APPROXIMATE NODAL VALUES AND ERRORS ***

NODE NUMBER	STATE VARIABLE	STREAM FUNCTION	ERROR
1	-.1837D-02	.3945D-01	-.1837D-02
2	.9825D 00	.4095D-01	.4095D-01
3	.1696D 01	.1824D-01	.1824D-01
4	.1956D 01	.7498D-02	.7498D-02
5	.2146D 01	.3564D-02	.3564D-02
6	.2511D 01	.1236D-01	.1140D-01
7	.2397D 01	.8037D 00	-.3115D-02
8	.2261D 01	.1793D 01	.1053D-01
9	.1193D 01	.1630D 01	-.6949D-02
10	-.4364D-02	.1508D 01	-.4364D-02
11	-.1744D-02	.8601D 00	-.1744D-02

*** BESSEL INEQUALITY ***
.4496D 02 >= .4495D 02 AND THE DIFFERENCE IS .1752D-02