1977

Software for Linear Elliptic Problems on General Two Dimensional Domains

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
77-220

SOFTWARE FOR LINEAR ELLIPTIC PROBLEMS ON GENERAL TWO DIMENSIONAL DOMAINS

E. N. Houstis* and J. R. Rice*
Mathematical Sciences
Purdue University

CSD-TR 220

February 1977

SOFTWARE FOR LINEAR ELLIPTIC PROBLEMS
ON GENERAL TWO DIMENSIONAL DOMAINS

E. N. Houstis* and J. R. Rice*
Mathematical Sciences
Purdue University

Abstract. In this paper we describe and illustrate software for the solution of a general elliptic partial differential equation defined on a general two dimensional domain and subject to mixed type boundary conditions.

1. INTRODUCTION

The numerical solution of elliptic partial differential equations can be a formidable programming task. Therefore, efficient, reliable, general well-documented software for solving such equations is needed. In this paper we describe a general purpose software ELLCOL for the solution of an elliptic partial differential equation

$$Lu \equiv \alpha u_{xx} + 2\beta u_{xy} + \gamma u_{yy} + \delta u_x + \varepsilon u_y + \zeta u = f$$

defined on a general two dimensional domain and subject to mixed type boundary conditions:

$$Bu \equiv au_x + bu_y + cu = g \quad \text{on } \partial\Omega \text{ boundary of } \Omega.$$

In section 2, we describe the modular structure of the software and the function of each module. In section 3, we provide some examples and the solutions computed from ELLCOL.

*Work supported in party by NSF grant MC376-10225

2. DESCRIPTION OF THE MODULES

I. Input Module

This is part of the "user interface" for the system (the other part is in Output modules). The user must specify:

a. Region boundary definition: user supplied routine

   SUBROUTINE BCOORD(P,X,Y)

   to define the boundary as a parametric curve; the X,Y values are returned corresponding to the parametric value P.

b. A rectangular mesh.

c. Problem definition: user supplied FORTRAN functions

   FUNCTION COEF(X,Y,J), J = 1,2,...,6 for α, β, ..., ζ

   FUNCTION BCOEF(X,Y,J), J = 1,2,3 for a,b,c

   which evaluate α, β, ..., ζ, a, b, c and

   FUNCTION F(X,Y,J), J = 1,2 for f,g

   that computes the functions f, g in the differential and boundary operator.

II. Region Module: Subroutine Region

This module processes the rectangular grid and the specified boundary. Its general objective is to locate the region with respect to the grid and to provide various informations useful in further processing. The output from this module consists of grid specifications and boundary specifications in the form of arrays which include coordinates of boundary intersection points, their type, and neighboring grid points and so on.

III. Equation Construction Module

This module consists of three components:

i) SUBROUTINE CLDATA which identifies the type of each element, calculates the number of boundary collocation points associated with each element, and calculates the boundary collocation points and the modified nodes of each element.

ii) SUBROUTINE INDEXING which numbers of the nodes of the final mesh to be used later for the approximation of the differential operator, determines the element incidences and estimates the bandwidth.

iii) SUBROUTINE FORMEQ which generates the system of the collocation equations. The bicubic Hermite rectangular elements are used to approximate the solution of the elliptic partial differential equation. We determine the approximate solution so that it satisfies the operator equation at four Gaussian points inside each element. The rest of the degrees of freedom of the approximate solution are determined by requiring it to approximately satisfy the boundary conditions. See reference [1]. FORMEQ stores the coefficient matrix and right side according to the output of indexing module in a band storage mode.

IV. Equation Solving Module: Subroutine BNDSOL

This module solves the system of collocation equations using profile Gaussian elimination.

V. Output Module

By selecting an output "level" the user can obtain

i) the output generated by the REGION module

ii) the input data to the subroutine FORMEQ

iii) the values of the approximate solution U and its derivatives U_x, U_y, U_{xy} evaluated at the nodes or at any set of points supplied by the user

iv) measurement of performance of each module in execution time.

3. NUMERICAL EXAMPLES

We describe three problems of various types which we feel will give an indication of the potential usefulness of our software.

A. We consider the numberical solution of the equation

$$(e^{xy}u_x)_x + (e^{-xy}u_y)_y - \frac{u}{1+x+y} = f \quad , \qquad \Omega = [0,1] \times [0,1]$$

subject to Dirichlet boundary conditions

$$u = 0.$$

In Table 1, we depict a typical output of the software ELLCOL for the solution of the above problem. See [1], [2] for other uses of this example.

B. Torsion of Compound Bars. The equation of torsion of a compound bar of cyclic cross section is

$$u_{xx} + u_{yy} = 0 \quad , \qquad \Omega = \text{circle with center at } (.5, .5) \text{ and radius } 1.$$

We assume the Neumann boundary conditions

$$u_N = g$$

where g is chosen such that $u = \tan^{-1}(y/x)$.

In Table 2, we show the solution and its derivatives found by ELLCOL at the nodes of a mesh generated by the REGION module.

C. Steady flow. The equation of a flow past a sphere is

$$u_{xx} + u_{yy} = f$$

defined on the domain Ω, see Figure 1 where we indicate the boundary conditions assumed on the graphical output from REGION. The function f has been determined such that

$$u = y[(x-2)^2 + y^2 - 1]e^{-.0625 \times (x-4)(y-2)}/[(3+(x-2)^2)(3+y^2)] \ .$$

Table 3 shows the solution obtained by ELLCOL. Finally, in Table 4 we present a version of the INPUT, OUTPUT, FORMEQ, and SOLVE modules.

REFERENCE

1. E. N. Houstis, R. E. Lynch, T. S. Papatheodorou, and J. R. Rice, Evaluation of numerical methods for Elliptic partial differential equations, submitted for publication in the Journal of Computational Physics.

2. Eisenstat, S.C. and M. H. Schultz [1973], Complexity of partial differential equations, in "Complexity of Sequential and Parallel Numerical Algorithms," (J. F. Traub, ed.), Academic Press, pp. 271-282.

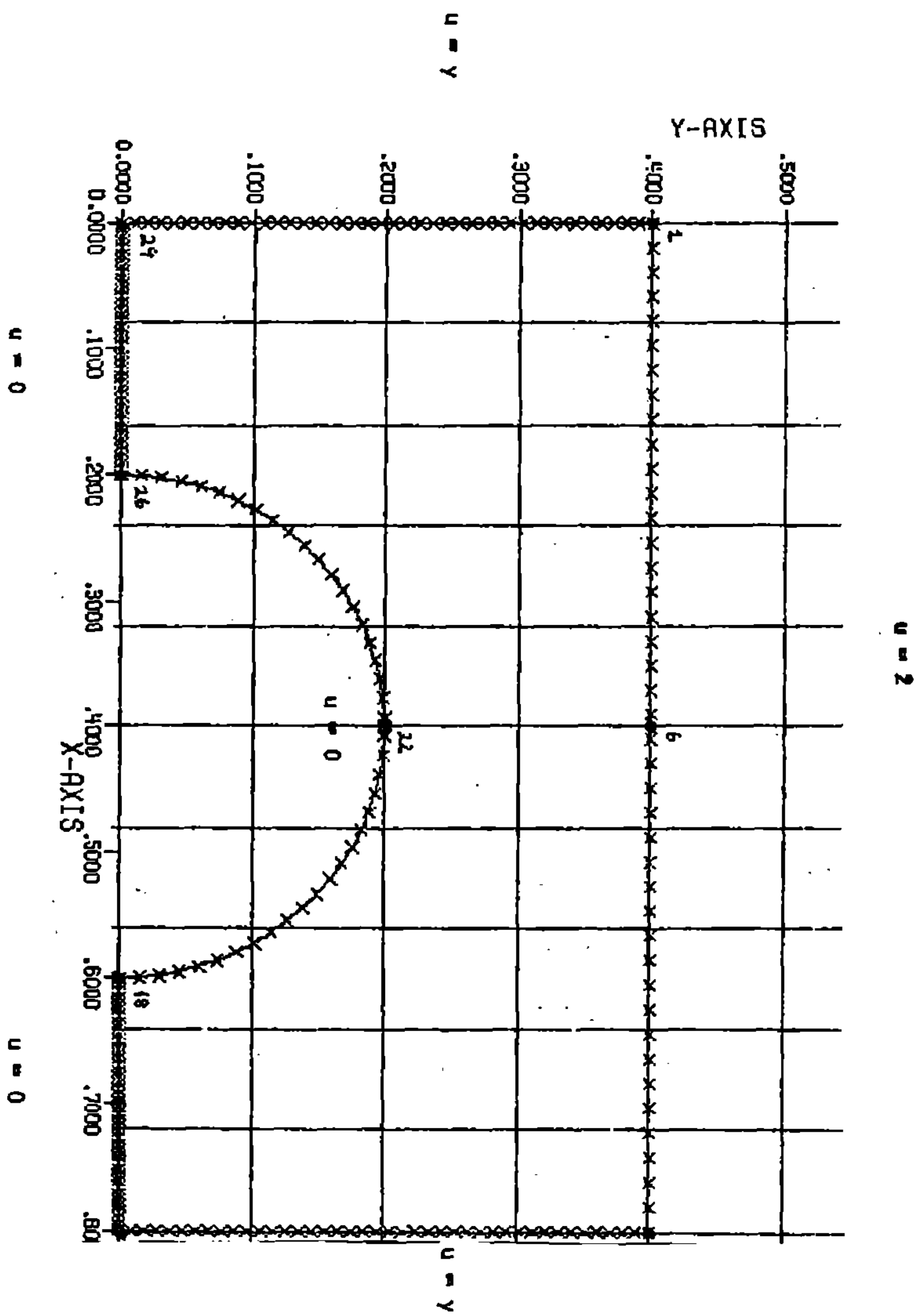

Figure 1. The domain of steady flow problem and the grid generated by REGION.

Table 1. Numerical solution of example A.

*** C O L L O C A T I O N P R O C E D U R E ***
T E S T P R O B L E M NO. 1

NODE	X-NODE COORDINATE	Y-NODE COORDINATE
1	-0	-0
2	0	.50000
3	0	1.00000
4	.50000	0
5	.50000	.50000
6	.50000	1.00000
7	1.00000	0
8	1.00000	.50000
9	1.00000	1.00000

MODIF NODE	X-NODE COORD	Y-COORDIN
1	-0	-0
2	0	.50000
3	0	1.00000
4	.50000	0
5	.50000	.50000
6	.50000	1.00000
7	1.00000	0
8	1.00000	.50000
9	1.00000	1.00000

ELEMENT	ELEMENT INCIDENCIES				NO. OF BOUNDARY COL. PTS	ELEMENT TYPE
1	1	4	5	2	5	
2	2	5	6	3	5	
3	4	7	8	5	5	
4	5	8	9	6	5	

BOUNDARY COLLOCATION POINTS

ELEMENT	X-COORD	Y-COORD	X-COORD	Y-COORD	X-COORD	Y-COORD
1	0	.16667	0	.33333	0	0
	.33333	0	.16667	0		
2	0	.66667	0	.83333	0	1.00000
	.33333	1.00000	.66667	1.00000		
3	1.00000	.33333	1.00000	.16667	1.00000	0
	.66667	0	.83333	0		
4	.66667	1.00000	.83333	1.00000	1.00000	1.00000
	1.00000	.66667	1.00000	.83333		

INPUT DATA TIME = .046

E Q U A T I O N P A R A M E T E R S
TOTAL NUMBER OF EQUATIONS = 36
BANDWIDTH = 20
O V E R A L L T I M E
TOTAL COEFFICIENT MATRIX FORMATION = .092
EQUATION SOLUTION = .313
TOTAL TIME FOR GENERATION AND SOL OF SYST .406

NODE	X-CO	Y-CO	TRUE SOLUTION	U	UX	UY	UXY	ABS. ERROR
1	-0	-0	0	-.000000	-.000000	.000000	9.983656	7.913E-14
2	0	.500	0	.000000	3.149958	.000000	-.105364	-3.988E-13
3	0	1.000	0	.000000	.000000	.000000	-9.659785	-1.791E-14
4	.500	0	0	.000000	-.000000	3.148165	-.122480	-7.681E-13
5	.500	.500	1.284025	1.295128	.629413	.643628	1.653215	-1.110E-02
6	.500	1.000	-.000000	.000000	-.000000	-5.210547	-5.279378	-1.647E-13
7	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

*** COLLOCATION PROCEDURE ***
TEST PROBLEM NO. 2

EQUATION PARAMETERS

TOTAL NUMBER OF EQUATIONS = 64
BANDWIDTH = 24

OVERALL TIME

TOTAL COEFFICIENT MATRIX FORMATION = .136
EQUATION SOLUTION = .897
TOTAL TIME FOR GENERATION AND SOL OF SYST 1.035

NODE	X-CO	Y-CO	TRUE SOLUTION	U
1	0	0	2.570000	1.786339
2	0	.333	2.570000	2.561287
3	0	.667	2.570000	2.573405
4	0	1.000	2.570000	2.573988
5	.333	0	1.000000	1.009670
6	.333	.333	1.785398	1.785483
7	.333	.667	2.107149	2.108117
8	.333	1.000	2.249046	2.250399
9	.667	0	1.000000	.997552
10	.667	.333	1.463648	1.462851
11	.667	.667	1.785398	1.785485
12	.667	1.000	1.982794	1.983398
13	1.000	0	1.000000	.996980
14	1.000	.333	1.321751	1.320580
15	1.000	.667	1.588003	1.597580
16	1.000	1.000	1.785398	1.784629

Table 2. Numerical solution of Tors

UX	UY	UXY	ABS. ERROR
-10.408614	10.381871	.255822	7.837E-01
-2.964479	.201596	7.005979	8.713E-03
-1.502434	.002251	2.187833	-3.405E-03
-1.016089	.013261	.729141	-3.988E-03
-.202144	2.963761	-6.990325	-9.670E-03
-1.505094	1.504998	.001192	-8.464E-05
-1.204113	.601652	1.078194	-9.682E-04
-.903369	.300760	.709917	-1.353E-03
-.002246	1.502410	-2.186753	2.448E-0[illegible]
-.601622	1.204143	-1.078194	7.967E-04
-.752268	.752173	-.001192	-8.689E-05
-.694638	.462321	.259015	-6.046E-04
-.013272	1.016077	-.729145	3.020E-03
-.300783	.903374	-.710997	1.171E-03
-.463039	.694090	-.274670	4.223E-04
-.513986	.487243	-.255826	7.691E-04

lon Compound Bar problem.

*** C O L L O C A T I O N   P R O C E D U R E ***

E Q U A T I O N   P A R A M E T E R S
TOTAL NUMBER OF EQUATIONS = 164
BANDWIDTH = 28

O V E R A L L   T I M E
TOTAL COEFFICIENT MATRIX FORMATION = .478
EQUATION SOLUTION = 2.976
TOTAL TIME FOR GENERATION AND SOL OF SYST 3.456

NODE	X-CO	Y-CO	TRUE SOLUTION	U	UX	UY	UXY	ABS. ERROR
1	-0	-0		-.000000	-.000100	4.000000	-20.186723	
2	-0	.125		.500000	-2.069750	4.000000	-11.516495	
3	-0	.235		.940000	-2.721114	4.000000	-.317633	
4	-0	.375		1.500000	-1.892666	4.000000	11.399364	
5	-0	.500		2.000000	.000000	4.000000	18.445931	
6	.125	-0		-.000013	-.000552	1.786259	-15.718675	
7	.125	.125		.261017	-1.787006	2.578704	-11.274303	
8	.125	.235		.613126	-2.535428	3.851376	-1.926814	
9	.125	.375		1.262407	-1.916104	5.361122	10.488363	
10	.125	.500		2.000000	.000000	6.398383	19.930133	
11	.260	-0		-.000144	-.005759	-.130055	-12.591212	
12	.260	.125		.035145	-1.535614	1.093013	-10.422782	
13	.260	.235		.283580	-2.295773	3.525599	-2.689615	
14	.260	.375		1.005994	-1.829096	6.715513	9.378329	
15	.260	.500		2.000000	.000000	9.126872	19.822496	
16	.375	.125		-.118998	-1.143923	.113191	-1.997506	
17	.375	.235		.054126	-1.574505	3.233602	-2.105478	
18	.375	.375		.821012	-1.281408	7.649998	6.391971	
19	.375	.500		2.000000	.000000	11.149548	14.120531	
20	.500	.235		-.050765	.000000	3.088553	-.000000	
21	.500	.375		.735303	.000000	8.075542	-.000000	
22	.500	.500		2.000000	.000000	12.096421	.000000	
23	.625	.125		-.118998	1.143923	.113191	1.997505	
24	.625	.235		.054126	1.574505	3.233602	2.105478	
25	.625	.375		.821012	1.281408	7.649998	-6.391971	
26	.625	.500		2.000000	.000000	11.149548	-14.120531	
27	.740	-0		-.000144	.005759	-.130055	12.591211	
28	.740	.125		.035145	1.535614	1.093013	10.422782	
29	.740	.235		.283580	2.295773	3.525599	2.689615	
30	.740	.375		1.005994	1.829096	6.715513	-9.378329	
31	.740	.500		2.000000	.000000	9.126872	-19.822496	
32	.875	-0		-.000013	.000552	1.786259	15.718675	
33	.875	.125		.261017	1.787006	2.578704	11.274303	
34	.875	.235		.613126	2.535428	3.851376	1.926814	
35	.875	.375		1.262407	1.916104	5.361122	-10.488363	
36	.875	.500		2.000000	.000000	6.398383	-19.930133	
37	1.000	-0		-.000000	.000100	4.000000	20.186723	
38	1.000	.125		.500000	2.069750	4.000000	11.516495	
39	1.000	.235		.940000	2.721114	4.000000	.317633	
40	1.000	.375		1.500000	1.892666	4.000000	-11.399364	
41	1.000	.500		2.000000	.000000	4.000000	-18.445931	

Table 3. Numerical solution of steady flow problem.

Table 4. INPUT, OUTPUT, FORMEQ, and SOLVE modules of software ELLCOL.

```
C      TEST  PROBLEM
       DIMENSION A(36,40),B(36)
C      A : COEFFICIENT MATRIX OF THE COLLOCATION EQUATIONS
C      B : INPUT THE RIGHT SIDE OF THE SYSTEM OF COLLOCATION
C          EQUATIONS
C          OUTPUT THE SOLUTION OF THE SYSTEM OF COLLOCATION
C          EQUATIONS
C          AT NODE ≠I≠    B(4*I-3) = U
C                         B(4*I-2) = UY
C                         B(4*I-1) = UX
C                         B(4*I)   = UXY
C      NDIMEN : NO. OF DEGREES OF FREEDOM = 4*NO. OF NODES
C      NCOLMN : 2*HALF BAND WIDTH OF A
C      NOTE : ONLY AN UPPER ESTIMATE NEEDED FOR -NCOLMN-
C      ----
C
C      COMPUTE THE COLLOCATION APPROXIMATION IN THE SPACE OF
C      PIECEWISE BICUBIC HERMITE POLYNOMIALS
C
       NDIMEN = 36
       NCOLMN = 40
C
       CALL COLLOC(A,B,NDIMEN,NCOLMN)
C
       STOP
       END
       SUBROUTINE COLLOC(A,B,NDIMEN,NCOLMN)
       DIMENSION A(NDIMEN,NCOLMN),B(NDIMEN)
       INTEGER TITLE(79)
C
       WRITE(6,10)
C
C      READ HEADING CARD
C
 21    READ(5,20) ICARD,TITLE
C
C      PRINT HEADING CARD
C
       WRITE(6,30) TITLE
       IF( ICARD .NE. 0 )                         GO TO 21
C      READ MASTER CONTROL CARD
       READ(5,31) INPUT,IEXACT,IDERIV
 31    FORMAT(3I1)
C
C      READ AND/OR PRINT INPUT DATA
C
       CALL MESH(INPUT,IDERIV)
C
       CALL SECOND(T0)
C
C      COMPUTE THE HALF BANDWIDTH OF THE COEFFICIENT MATRIX
C
       CALL BDWDTH(NBWDTH)
C
       NCODIG = 2*NBWDTH - 1
C
C
C      PRINT THE COLLOCATION EQUATIONS PARAMETERS
C
       WRITE(6,40)
       WRITE(6,50) NDIMEN
       WRITE(6,60) NBWDTH
       WRITE(6,70)
C
C      SOLVE THE LINEAR SYSTEM OF COLLOCATION EQTS.
C
       CALL SOLVE(A,B,NDIMEN,NCODIG,NBWDTH)
C
C      COMPUTE APPROXIMATE SOLUTION U ,UX ,UY , UXY
C
```

```
      CALL DERIV(B,NDIMEN,IEXACT)
C
C     EVALUATE THE EXACT SOLUTION (IF ANY),
C     APPROXIMATE SOLUTION,MAXIMUM ERROR,RELATIVE ERROR
C     AT A GIVEN SET OF POINTS
C
      CALL SUMARY(IEXACT,B,NDIMEN)
C
      CALL SECOND(T1)
C
C     MEASURE THE TOTAL COLLOCATION SOLUTION TIME
C
      EXTIME = T1 - T0
      WRITE(6,80) EXTIME
 10   FORMAT(1H1,10X,30H*** C O L L O C A T I O N  P R
     A              ,17HO C E D U R E ***)
 20   FORMAT(I1,79A1)
 30   FORMAT(1H0,10X,80A1)
 40   FORMAT(1H0,36HE Q U A T I O N  P A R A M E T E R S)
 50   FORMAT(1H0,25HTOTAL NUMBER OF EQUATIONS,15X,1H=,I5)
 60   FORMAT(1X,9HBANDWIDTH,31X,1H=,I5)
 70   FORMAT(1H0,22HO V E R A L L  T I M E)
 80   FORMAT(1H0,31HTOTAL COLLOCATION SOLUTION TIME,8X,1H=,F6.3)
                                                       RETURN
      END
      SUBROUTINE DERIV(B,NDIMEN,IEXACT)
      COMMON /NODCOR/ XNODE(100) ,YNODE(100) ,NNODES
      DIMENSION B(NDIMEN)
C
C     OUTPUTS THE TRUE SOLUTION, U , UX , UY , UXY AT THE NODES
C
      WRITE(6,130)
      WRITE(6,100)
      WRITE(6,130)
      DO 110 NOD = 1 , NNODES
         XPNT = XNODE(NOD)
         YPNT = YNODE(NOD)
         IF( IEXACT .EQ. 1 ) TRSOL = TRUE(XPNT,YPNT)
C
C     COMPUTE THE INDECIES CORRESPONDING TO DEGREES OF FREEDOM
C     PER NODE
C
         NDF1 = 1 + (NOD-1)*4
         NDF2 = NDF1 + 1
         NDF3 = NDF2 + 1
         NDF4 = NDF3 + 1
C
C     COMPUTE U ,UX ,UY ,UXY
C
         U   = B(NDF1)
         UY  = B(NDF2)
         UX  = B(NDF3)
         UXY = B(NDF4)
C
         IF( IEXACT .EQ. 1) ERROR = TRSOL - U
         IF( IEXACT .EQ. 1) WRITE(6,120) NOD,XPNT,YPNT,TRSOL,U,UX,UY,UXY
     A                                       ,ERROR
         IF( IEXACT .EQ. 0) WRITE(6,121) NOD,XPNT,YPNT,U,UX,UY,UXY
C
 110   CONTINUE
 100  FORMAT(1X,4HNODE,6X,4HX-CO,6X,4HY-CO,2X,13HTRUE SOLUTION
     A          ,14X,1HU,13X,2HUX,13X,2HUY,12X,3HUXY,4X,
     B           11HABS.  ERROR)
 120   FORMAT(1X,I4,2F10.3,5F15.6,E15.3)
 130   FORMAT(115(1H-))
 121   FORMAT(1X,I4,2F10.3,15X,4F15.6)
       RETURN
       END
      SUBROUTINE MESH(INPUT)
C
C     READS AND/OR OUTPUTS THE MESH SPECIFICATIONS AND
C     BOUNDARY COLLOCATION POINTS
```

```
C
      COMMON /NODCOR/ XNODE(100) ,YNODE(100) ,NNODES
      COMMON /ELINCD/ LOLFT(100) ,LORGT(100) ,
     A                UPRGT(100) ,UPLFT(100) ,NUMEL
      COMMON /ELTYPE/ TYPE(100)
      COMMON /BNDCOL/ NBOCO(100) ,BOX(100,10) ,BOY(100,10)
      COMMON /MODNOD/ XBNODE(100) ,YBNODE(100)
C
      INTEGER UR ,UL ,TYP ,TYPE ,UPLFT , UPRGT
C
      CALL SECOND(T0)
C
C     READ NODAL POINT COORDINATES
C
 30   READ(5,10) NOD ,XNOD ,YNOD
      IF( NOD .EQ. 0     )                  GO TO 20
          XNODE(NOD) = XNOD
          YNODE(NOD) = YNOD
C
          XBNODE(NOD) = XNOD
          YBNODE(NOD) = YNOD
C
          NNODES = NOD
                                            GO TO 30
C
C     READ THE COORDINATES OF THE MODIFIED NODES
C
 20   READ(5,40) NOD ,XBNOD ,YBNOD
      IF( NOD .EQ. 0     )                  GO TO 50
          XBNODE(NOD) = XBNOD
          YBNODE(NOD) = YBNOD
                                            GO TO 20
C
C     READ ELEMENT INCIDENCIES ,ELEMENT TYPE AND NUMBER OF
C     BOUNDARY COLLOCATION POINTS CORRESPONDING TO EACH ELEMENT
C
 50   READ(5,60) NEL,LL,LR,UR,UL,NBCPT,TYP
      IF( NEL .EQ. 0       )                GO TO 70
        LOLFT(NEL) = LL
        LORGT(NEL) = LR
        UPRGT(NEL) = UR
        UPLFT(NEL) = UL
        NBOCO(NEL) = NBCPT
        TYPE(NEL)  = TYP
        NUMEL      = NEL
                                            GO TO 50
C
C     READ BOUNDARY COLLOCATION POINTS

 70   DO 80 NEL = 1 , NUMEL
         IF( NBOCO(NEL) .EQ. 0 )            GO TO 80
           NDUM = NBOCO(NEL)
           READ(5,100) NDUM,(BOX(NEL,NPT),BOY(NEL,NPT),NPT=1,NDUM)
 80   CONTINUE
      CALL SECOND(T1)
      IF( INPUT .EQ. 0 )                    GO TO 110
C
C     OUTPUT THE INPUT DATA
C
C
      WRITE(6,120)
      WRITE(6,130)
      WRITE(6,120)
      DO 140 NODE = 1 , NNODES
      WRITE(6,150) NODE, XNODE(NODE) ,YNODE(NODE)
 140  CONTINUE
      WRITE(6,120)
C
      WRITE(6,120)
      WRITE(6,160)
      WRITE(6,120)
      DO 170 NODE = 1 , NNODES
```

```
         WRITE(6,150) NODE , XBNODE(NODE) ,YBNODE(NODE)
 170  CONTINUE
      WRITE(6,120)
      WRITE(6,120)
      WRITE(6,180)
      WRITE(6,120)
      DO 190 NEL = 1 , NUMEL
         WRITE(6,200) NEL,LOLFT(NEL),LORGT(NEL),UPRGT(NEL),
     A                       UPLFT(NEL),NBOCO(NEL),TYPE(NEL)
 190  CONTINUE
C
C     OUTPUT THE BOUNDARY COLLOCATION POINTS
C
      WRITE(6,120)
      WRITE(6,191)
      WRITE(6,120)
      WRITE(6,192)
      DO 210 NEL = 1 , NUMEL
         IF( NBOCO(NEL) .EQ. 0 ) GO TO 210
            KDUM = NBOCO(NEL)
         WRITE(6,240) NEL,(BOX(NEL,J),BOY(NEL,J),J=1,KDUM)
 210  CONTINUE
      WRITE(6,120)
      EXTIME = T1 - T0
      WRITE(6,250) EXTIME
      WRITE(6,120)
 10   FORMAT(I5,2F10.0)
 40   FORMAT(I5,2F10.0)
 60   FORMAT(6I5,I2)
 100  FORMAT(I5,6F10.0/(6F10.0))
 101  FORMAT(I5,6F10.0/(5X,6F10.0))
 120  FORMAT(1X,80(1H-))
 130  FORMAT(1X,48HNODE      X-NODE COORDINATE      Y-NODE COORDINATE)
 150  FORMAT(5X,I4,7X,F10.5,12X,F10.5)
 160  FORMAT(5X,41HMODIF NODE      X-NODE COORD      Y-COORDIN)
 180  FORMAT(6X,7HELEMENT,13X,19HELEMENT INCIDENCIES
     A        ,3X,24HNO. OF BOUNDARY COL. PTS
     B        ,3X,12HELEMENT TYPE)
 200  FORMAT(1X,I9,13X,4I5,7X,15X,I5,10X,I1)
 240  FORMAT(1X,I5,4X,6F10.5/(10X,6F10.5))
 191  FORMAT(1X,27HBOUNDARY COLLOCATION POINTS)
 192  FORMAT(1X,7HELEMENT,3(5X,7HX-COORD,3X,7HY-COORD))
 250  FORMAT(5X,15HINPUT DATA TIME,25X,1H=,F6.3)
 110  RETURN
      END
      SUBROUTINE SUMARY(IEXACT,B,NDIMEN)

C  MEASURES THE PERFORMANCE OF THE COLLOCATION  PROCEDURE
      DIMENSION B(NDIMEN)
      INTEGER IEXACT
C  INITIALIZATIONS
      ERMAX = -1.
      NCARD = 0
      RELMER = -10000.
      WRITE(6,1)
C  READ THE COORDINATES OF THE POINTS IN THE GIVEN
C  DOMAIN OF DEFINITION,AT WHICH THE SOLUTION IS COMPUTED
C
      WRITE(6,10)
 10   FORMAT(1H0,14HX-CORD  Y-CORD,2X,13HTRUE SOLUTION,1X
     A           ,16HAPPROX. SOLUTION,1X,14HABSOLUTE ERROR,1X
     B           ,14HRELATIVE ERROR)
      WRITE(6,1)
 30   READ(5,20) XPOINT,YPOINT
      NCARD = NCARD + 1
      IF( XPOINT .EQ. -9000. )                              GO TO 40

C  COMPUTE THE TRUE SOLUTION AT (XPOINT,YPOINT)
C
      IF( IEXACT .EQ. 1 ) TRSOL = TRUE(XPOINT,YPOINT)
         CALL APPR(APRSOL,XPOINT,YPOINT,B,NDIMEN)
C  COMPUTE THE MAXIMUM ERROR
```

```
      IF( IEXACT .EQ. 0 )                                    GO TO 51
              ERROR = ABS( TRSOL - APRSOL )
              IF( ERROR .GT. ERMAX ) ERMAX = ERROR
C
C  COMPUTE THE RELATIVE ERROR
C
         RELERR = ERROR / TRSOL
         IF( RELERR .GT. RELMER )    RELMER = RELERR
      WRITE(6,60 ) XPOINT,YPOINT,TRSOL,APRSOL,ERROR,RELERR
                                                             GO TO 30
 51   WRITE(6,61)   XPOINT,YPOINT,APRSOL
 61   FORMAT(1X,F5.2,3X,F5.2,2X,F15.7)
 60   FORMAT(1X,F5.2,3X,F5.2,2X,2G15.7,2G15.3)
                                                            GO TO 30
 40   IF( IEXACT .EQ. 0 )                                    RETURN
      IF( NCARD .EQ. 1)                                      RETURN
      WRITE(6,70 ) ERMAX,RELMER
 100  FORMAT(2F10.0)
 70   FORMAT(1H0,21HMAXIMUM ABS. ERROR = ,E15.5,5X,
     A           21HMAXIMUM REL. ERROR = ,E15.5)
 20   FORMAT(2F10.0)
  1   FORMAT(80(1H*))
      RETURN
      END
      SUBROUTINE SOLVE(A,B,NDIMEN,NCODIG,NBWDTH)
C
C  SOLVES THE SYSTEM OF COLLOCATION EQUATIONS
C
      DIMENSION A(NDIMEN,NCODIG),B(NDIMEN)
      COMMON/NODCOR/XNODE(100),YNODE(100),NNODES
C
          CALL SECOND(T01)
C
C   FORM THE COLLOCATION EQUATIONS
C
          CALL EQUATE(A,B,NDIMEN,NCODIG,NBWDTH)
C
          NDUM = NBWDTH - 1
C
          CALL SECOND(T0)
C
C
C   SOLVE THE COLLOCATION EQUATIONS
C
          CALL BNDSOL(NDIMEN,NDUM,A,B)
C
          CALL SECOND(T1)
C
C
C   MEASURE THE GENERATION AND SOLUTION EQUATIONS TIME
C
          EXTIME = T1 - T0
          WRITE(6,100) EXTIME
 100      FORMAT(1X,17HEQUATION SOLUTION,23X,1H=,F6.3)
          TOTEX = T1 - T01
          WRITE(6,110) TOTEX
 110      FORMAT(1X,41HTOTAL TIME FOR GENERATION AND SOL OF SYST,F6.3)
                                                  RETURN
      END
      SUBROUTINE BASE(B1,B2,B3,B4,Z,H)
C
C    THIS ROUTINE DEFINES THE CUBIC HERMITE POLYNOMIALS
C
      S=Z/H
      T=1.-S
      B1=S*S*(2.*S-3.)+1.
      B2=1.-B1
      B3=S*T*(H-Z)
      B4=-S*T*Z
      RETURN
      END
```

```
      SUBROUTINE DBASE(DB1,DB2,DB3,DB4,Z,H)
C
C       THIS ROUTINE DEFINES THE FIRST DERIVATIVES OF
C       THE CUBIC HERMITE POLYNOMIALS
C
      RH = 1./H
      S = Z*RH
      T=1.-S
      DB1 = 6.*(S - 1.)*S*RH
      DB2=-DB1
      DB3 = T*(T - 2.*S)
      DB4 = S*(S - 2.*T)
      RETURN
      END
      SUBROUTINE DDBASE(DDB1,DDB2,DDB3,DDB4,Z,H)
C
C       THIS ROUTINE DEFINES THE  SECOND  DERIVATIVES OF
C       THE CUBIC HERMITE POLYNOMIALS
C
      RH = 1./H
      S = Z*RH
      T=1.-S
      DDB1 = 6.*(2.*S - 1.)*RH*RH
      DDB2=-DDB1
      DDB3 = 2.*(S - 2.*T)*RH
      DDB4 = 6.*(S - T)*RH - DDB3
      RETURN
      END
      SUBROUTINE BNDSOL(N,M,A,B)
      DIMENSION A(N,M),B(N)
C
C       L-U DECOMPOSITION OF AN UNSYMMETRIC BAND MATRIX A
C       HAVING BAND WIDTH M=MAX/I-J/ FOR A(I,J)=0.
C       THE DIAGONALS OF THE MATRIX ARE ASSUMED TO BE
C       STORED IN THE COLUMNS OF A.
C
      M1=M+1
      M2=M+2
      MM1=M+M+1
C
      L=M
      DO 3 I=1,M
      K=M2-I
      DO 4 J=K,MM1
      IDUM=J-L
      A(I,IDUM)=A(I,J)
    4 CONTINUE
      L=L-1
      K=MM1-L
      DO 5 J=K,MM1
      A(I,J)=0.
    5 CONTINUE
    3 CONTINUE
C
C       END INITIALIZATION
C
      L=M
      DO 6 K=1,N
      X=ABS(A(K,1))
      I=K
      IF(L.LT.N) L=L+1
      KP1=K+1
      IF(KP1.GT.L) GO TO 12
C
C       SEARCH FOR INTERCHANGE
C
      DO 7 J=KP1,L
      IF(ABS(A(J,1)).LE.X) GO TO 7
      X=ABS(A(J,1))
      I=J
    7 CONTINUE
   12   IF(I.EQ.K) GO TO 8
```

```
C
C       INTERCHANGE ROWS I AND K
C
      X = B(K)
      B(K) = B(I)
      B(I) = X
      DO 9 J=1,MM1
      X=A(K,J)
      A(K,J)=A(I,J)
      A(I,J)=X
    9 CONTINUE
C
    8 IF (KP1.GT.L) GO TO 6
      RAK1 = 1./A(K,1)
      DO 10 I=KP1,L
      IF ( A(I,1) .EQ. 0. )                          GO TO 13
           X = A(I,1) * RAK1
      DO 11 J = 2 , MM1
      A(I,J-1) = A(I,J) - X * A(K,J)
   11 CONTINUE
      B(I) = B(I) - X*B(K)
                                                     GO TO 15
   13 CONTINUE
      DO 14 J = 2 , MM1
      A(I,J-1) = A(I,J)
   14 CONTINUE
   15 A(I,MM1) = 0.
   10 CONTINUE
    6 CONTINUE
C
C       SOLVE A.X=B AND PLACE THE RESULT IN B.
C
      L=1
      DO 40 J=1,N
      I=N+1-J
      IM1=I-1
      X=B(I)
      IF(I.EQ.N) GO TO 50
      IF(L.LT.MM1) L=L+1
      DO 60 K=2,L
      KDUM=IM1+K
      X=X-A(I,K)*B(KDUM)
   60 CONTINUE
   50 B(I)=X/A(I,1)
   40 CONTINUE
      RETURN
      END
      SUBROUTINE APPR(APRSOL,XP,YP,B,NDIMEN)
      DIMENSION B(NDIMEN)
      COMMON/NODCOR/XNODE(100),YNODE(100),NNODES
      COMMON/ELINCD/LOLFT(100),LORGT(100),UPRGT(100),UPLFT(100)
     A              ,NUMEL
      INTEGER UL,UPRGT,UPLFT
C
C       IT RETURNS THE VALUE OF COLLOCATION
C       OXIMATION EVALUATED AT (X,Y) POINT
C
C       FINDS THE ELEMENT THAT (X,Y) BELONGS.
C
      DO 101 NEL=1,NUMEL
      LL=LOLFT(NEL)
      LR=LORGT(NEL)
      IF(XNODE(LL).LE.XP.AND.XP.LE.XNODE(LR)) GO TO 102
      GO TO 101
  102 UL=UPLFT(NEL)
      IF(YNODE(LL).LE.YP.AND.YP.LE.YNODE(UL)) GO TO 104
  101 CONTINUE
C
C       COMPUTES DIMENSIONS OF NEL ELEMENT FOUND
C
  104 HX = XNODE(LR) - XNODE(LL)
      HY = YNODE(UL) - YNODE(LL)
```

```
C     COMPUTES THE BASES FUNCTIONS AT (X,Y)PT.
C
      X = XP - XNODE(LL)
      Y = YP - YNODE(LL)
C
C    COMPUTE BASES FUNCTIONS AT (X,Y)
C
      CALL BASE(BASEX1,BASEX2,BASEX3,BASEX4,X,HX)
      CALL BASE(BASEY1,BASEY2,BASEY3,BASEY4,Y,HY)
C
          I1=4*LOLFT(NEL)-3
          I2=I1+1
          I3=I2+1
          I4=I3+1
          TEMP1 = B(I1)*BASEX1*BASEY1 + B(I2)*BASEX1*BASEY3
     A            + B(I3)*BASEX3*BASEY1 + B(I4)*BASEX3*BASEY3
          J1=4*LORGT(NEL)-3
          J2=J1+1
          J3=J2+1
          J4=J3+1
          TEMP2 = B(J1)*BASEX2*BASEY1 + B(J2)*BASEX2*BASEY3
     A            + B(J3)*BASEX4*BASEY1 + B(J4)*BASEX4*BASEY3
          K1=4*UPRGT(NEL)-3
          K2=K1+1
          K3=K2+1
          K4=K3+1
          TEMP3 = B(K1)*BASEX2*BASEY2 + B(K2)*BASEX2*BASEY4
     A            + B(K3)*BASEX4*BASEY2 + B(K4)*BASEX4*BASEY4
          L1=4*UPLFT(NEL)-3
          L2=L1+1
          L3=L2+1
          L4=L3+1
          TEMP4 = B(L1)*BASEX1*BASEY2 + B(L2)*BASEX1*BASEY4
     A            + B(L3)*BASEX3*BASEY2 + B(L4)*BASEX3*BASEY4
C
          APRSOL=TEMP1+TEMP2+TEMP3+TEMP4
      RETURN
      END
C
      SUBROUTINE EQUATE(A,B,NROWS,NCOLUM,NBWDTH)
C
C
C     GENERATES THE MATRIX A AND THE RIGHT SIDE VECTOR B
C     OF THE SYSTEM OF COLLOCATION EQUATIONS.
C
C     THE COLLOCATION APPROXIMATION UC(X,Y) IN THE SPACE OF
C     BICUBIC HERMITE PIECEWISE POLYNOMIALS RESTRICTED TO ELEMENT
C     ≠ NEL ≠ IS
C       UC(X,Y) = V(I1)BX1BY1+V(I2)BX1BY3+V(I3)BX3BY1+V(I4)BX3BY3+
C                 V(J1)BX2BY1+V(J2)BX2BY3+V(J3)BX4BY1+V(J4)BX4BY3+
C                 V(K1)BX2BY2+V(K2)BX2BY4+V(K3)BX4BY2+V(K4)BX4BY4+
C                 V(L1)BX1BY2+V(L2)BX1BY2+V(L3)BX3BY2+V(L4)BX3BY4
C     ACCORDING TO COLLOCATION PROCEDURE , UC(X,Y). SATISFIES
C
C          (I)       L(UC(T,S)) = F(T,S,1)
C
C     AT INTERIOR COLLOCATION POINTS
C
C     AND
C
C          (II)      B(UC(T,S)) = F(T,S,2)
C
C     AT THE BOUNDARY COLLOCATION POINTS WHERE
C       L(U(X,Y))=COEF(X,Y,1)*DDXU(X,Y)+COEF(X,Y,2)*DDYU(X,Y)+
C                 COEF(X,Y,3)*DXU(X,Y) +COEF(X,Y,4)*DYU(X,Y)+
C                 COEF(X,Y,5)*U(X,Y)
C
C       B(U(X,Y))=BCOEF(X,Y,1)*DXU(X,Y)+BCOEF(X,Y,2)*DYU(X,Y)+
C                 BCOEF(X,Y,3)*U(X,Y)
C
C     THE PROCEDURE THAT GENERATES A X = B FROM (I) , (II)
C     IS THE  FOLLOWING:
```

```
C
C      1. SET    A = 0
C
C      2. FOR EACH ELEMENT ≠ NEL ≠ IT COMPUTES THE COORDINATES
C         OF ITS FOUR INTERIOR COLLOCATION POINTS.
C
C      3. COMPUTE THE TOTAL NUMBER OF COLLOCATION POINTS
C         ASSOCIATED WITH ELEMENT ≠ NEL ≠
C
C      4. AT EACH COLLOCATION POINT OF ELEMENT ≠ NEL ≠ IT COMPUTES
C         THE BASIS FUNCTIONS ,OF THE HERMITE CUBIC PIECEWISE
C         POLYNOMIALS SPACE AND THEIR FIRST AND SECOND DERIVATIVES.
C
C      5. IF THE COLLOCATION POINT IS INTERIOR THEN IT GENERATES
C         COLLOCATION EQUATIONS BY SATISFYING EQUATION (I) AT THAT
C         POINT
C
C      6. IF THE COLLOCATION POINT IS BOUNDARY THEN IT GENERATES
C         A COLLOCATION EQUATION BY SATISFYING EQUATION (II) AT THAT
C         POINT.
C
C      7. IT MEASURES THE TIME USED TO GENERATE THE WHOLE MATRIX A
C         AND THE RIGHT HAND SIDE B.
C
C   V A R I A B L E   L I S T
C
C      A(NROWS,NCOLUM) :THE COEFFICIENT MATRIX OF THE SYSTEM OF
C                       COLLOCATION EQUATIONS
C      NROWS      :   THE TOTAL NUMBER OF ROWS
C      NCOLUM     :   THE TOTAL NUMBER OF COLUMNS
C      NR         :   ROW INDEX
C      NC         :   COLUMN INDEX
C      B(NROWS)   :   THE RIGHT HAND SIDE OF THE SYSTEM OF COLLOCATION
C                     EQUATIONS
C      NUMEL      :   TOTAL NUMBER OF ELEMENTS
C      LL,LR,UR,UL :  INTEGER IDENTIFIERS OF THE ELEMENT NODES
C      HX,HY      :   THE SIZES OF AN ELEMENT
C      XNODE( ),YNODE( ):THE COORDINATES OF THE ELEMENT NODES
C      X(1),Y(1),X(2),Y(2),...,X(4),Y(4):COORDINATES OF THE
C                      THE INTERIOR COLLOCATION POINTS
C      NTOTAL     :    NUMBER OF COLLOCATION POINTS ASSOCIATED
C                      WITH ELEMENT ≠ NEL≠
C      NBOCO( )   :    NUMBER OF BOUNDARY COLLOCATION PTS. ASSOCIATED
C                      WITH ELEMENT ≠ NEL ≠
C      BOX( , )   :    THE COORDINATES OF THE BOUNDARY COLLOCATION PTS.
C      BOY( , )
C      BX1,BX2,BX3,BX4 : THE VALUES OF BASES FUNCTIONS EVALUATED
C      BY1,BY2,BY3,BY4 : AT XDUM , YDUM RESPECTIVELY
C      DBX1,DBX2,DBX3,DBX4 :THE DERIVATIVES OF BASES FUNCTIONS
C      DBY1,DBY2,DBY3,DBY4 :EVALUATED AT XDUM,YDUM RESPECTIVELY
C      DDBX1,DDBX2,DDBX3,DDBX4:THE SECOND DERIVATIVES OF BASES
C      DDBY1,DDBY2,DDBY3,DDBY4:FUNCTIONS EVALUATED AT XDUM,YDUM
C                               RESPECTIVELY
C      COEF(X,Y,J),J=1,5 : THE COEFFICIENTS OF THE OPERATOR
C      BCOEF(X,Y,J),J=1,3:THE COEFFICIENTS OF THE BOUNDARY CONDITIONSS
C      F(X,Y,J) ,J=1,2 :THE RIGHT HAND SIDE OF THE OPERATOR AND
C                       BOUNDARY CONDITION
C
      DIMENSION A(NROWS,NCOLUM),B(NROWS),X(4),Y(4)
      COMMON/NODCOR/XNODE(100),YNODE(100),NNODES
      COMMON/ELINCD/LOLFT(100),LORGT(100),UPRGT(100),
     *              UPLFT(100),NUMEL
      COMMON/ELTYPE/TYPE(100)
      COMMON/BNDCOL/NBOCO(100),BOX(100,10),BOY(100,10)
      COMMON/MODNOD/XBNODE(100),YBNODE(100)
      DATA BOUND , INTER / 1 , 0 /
      INTEGER BOUND,TYPE,INTER,UPRGT,UPLFT,UR,UL
      CALL SECOND(T0)
C
C  SET   A = 0
C
      DO 100 NR = 1 , NROWS
```

```
      DO 100 NC = 1 , NCOLUM
         A(NR,NC) = 0.
 100  CONTINUE
C
C  INITIALIZE ROW INDEX
C
      NROW = 0
C
      INCREM = NBWDTH - 3
      SQRT3 = SQRT(3.)
      RSIX = 1./6.
      FACTOR = .5/SQRT3
C
C  COMPUTE THE ENTRIES OF MATRIX A AND RIGHT HAND SIDE B
C
      DO 110 NEL = 1 , NUMEL
C
C  ASSIGN TO VARIABLES LL,LR,UR,UL THE NODE NUMBERS OF ELEMENT ≠NEL≠
C
         LL = LOLFT(NEL)
         LR = LORGT(NEL)
         UR = UPRGT(NEL)
         UL = UPLFT(NEL)
C
C  COMPUTE THE ≠SIZES≠ OF ELEMENT ≠NEL≠
C
         HX = XNODE(LR) - XNODE(LL)
         HY = YNODE(UL) - YNODE(LL)
C
C  DETERMINE THE TYPE OF ELEMENT ≠NEL≠
C
         IF ( TYPE(NEL) .NE. INTER )                    GO TO 120
C
C  TYPE OF ELEMENT ≠NEL≠ IS ≠INTERIOR≠
C  COMPUTE ITS INTERIOR COLLOCATION POINTS.
C
             XMIDL = .5*(XNODE(LL) + XNODE(LR))
             YMIDL = .5*(YNODE(LL) + YNODE(UL))
             XFACTR = HX * FACTOR
             YFACTR = HY * FACTOR
C
C  COORDINATES OF INTERIOR COLLOCATION PTS.
C
             X(1) = XMIDL - XFACTR
             X(2) = XMIDL + XFACTR
             X(3) = X(2)
             X(4) = X(1)
             Y(1) = YMIDL - YFACTR
             Y(2) = Y(1)
             Y(3) = YMIDL + YFACTR
             Y(4) = Y(3)
                                                      GO TO 130
 120  CONTINUE
C
C  THE TYPE OF ELEMENT ≠NEL≠ IS  ≠BOUNDARY≠
C  COMPUTE ITS INTERIOR COLLOCATION PTS FOR THE
C  MODIFIED BOUNDARY ELEMENT ≠NEL≠
C
             XDUMY1 = XBNODE(LR) + XBNODE(UL)
             XDUMY2 = XBNODE(LL) + XBNODE(UR)
             YDUMY1 = YBNODE(LR) + YBNODE(UL)
             YDUMY2 = YBNODE(LL) + YBNODE(UR)
             XSUM = XDUMY1 + 2.*XDUMY2
             YSUM = YDUMY1 + 2.*YDUMY2
             XDIF = (XBNODE(UR) - XBNODE(LL))*SQRT3
             YDIF = (YBNODE(UR) - YBNODE(LL))*SQRT3
C
             X(1) = (XSUM - XDIF) * RSIX
             Y(1) = (YSUM - YDIF) * RSIX
             X(3) = (XSUM + XDIF) * RSIX
             Y(3) = (YSUM + YDIF) * RSIX
C
```

```
            XSUM = 2. * XDUMY1 + XDUMY2
            YSUM = 2. * YDUMY1 + YDUMY2
            XDIF = (XBNODE(UL) - XBNODE(LR)) * SQRT3
            YDIF = (YBNODE(UL) - YBNODE(LR)) * SQRT3
C
            X(2) = (XSUM - XDIF) * RSIX
            Y(2) = (YSUM - YDIF) * RSIX
            X(4) = (XSUM + XDIF) * RSIX
            Y(4) = (YSUM + YDIF) * RSIX
 130   CONTINUE
C
C   COMPUTE THE TOTAL NUMBER OF COLLOCATION PTS ASSOCIATED
C   WITH ELEMENT ≠NEL≠
C
            NTOTAL = 4 + NBOCO(NEL)
C
C
       DO 140 IDUM = 1 , NTOTAL
          IF ( IDUM .GT. 4 )                       GO TO 150
C
C   AN INTERIOR COLLOCATION PT OF ELEMENT ≠NEL≠
C
               T = X(IDUM)
               S = Y(IDUM)
                                                 GO TO 160
 150   CONTINUE
C
C   A BOUNDARY COLLOCATION POINT OF ELEMENT ≠NEL≠
C
               JDUM = IDUM - 4
               T = BOX(NEL,JDUM)
               S = BOY(NEL,JDUM)
C
 160   CONTINUE
C
C   TRANSFER ELEMENT ≠NEL≠ TO ELEMENT WITH NODES (0,0),
C   (HX,0),(HX,HY),(0,HY)
C
               XDUM = T - XNODE(LL)
               YDUM = S - YNODE(LL)
C
C   EVALUATE THE BASIS FUNCTIONS AND ITS DARIVATIVES
C   AT (XDUM,YDUM)
C
               CALL BASE(BX1,BX2,BX3,BX4,XDUM,HX)
               CALL BASE(BY1,BY2,BY3,BY4,YDUM,HY)
C
               CALL DBASE(DBX1,DBX2,DBX3,DBX4,XDUM,HX)
               CALL DBASE(DBY1,DBY2,DBY3,DBY4,YDUM,HY)
C
               CALL DDBASE(DDBX1,DDBX2,DDBX3,DDBX4,XDUM,HX)
               CALL DDBASE(DDBY1,DDBY2,DDBY3,DDBY4,YDUM,HY)
C
C   COMPUTE THE ROW INDEX
               NROW = NROW + 1
C
C   COMPUTE THE INDECIES OF THE UNKNOWNS V(.) ASSOCIATED
C   WITH THE ELEMENT ≠NEL≠.
C
               I1 = 4 * LL + INCREM - NROW
               I2 = I1 + 1
               I3 = I2 + 1
               I4 = I3 + 1
C
               J1 = 4 * LR + INCREM - NROW
               J2 = J1 + 1
               J3 = J2 + 1
               J4 = J3 + 1
C
               K1 = 4 * UR + INCREM - NROW
               K2 = K1 + 1
               K3 = K2 + 1
```

```
              K4 = K3 + 1
C
              L1 = 4 * UL + INCREM - NROW
              L2 = L1 + 1
              L3 = L2 + 1
              L4 = L3 + 1
C
C     FOR   IDUM .LE. 4 WE COMPUTE THE COEFFICIENTS OF THE UNKNOWNS IN
C     EQUATION (I)
C
              IF ( IDUM .GT. 4 )                 GO TO 170
C
C     EVALUATE THE COEFFICIENTS OF THE DIFFERENTIAL OPERATOR.
C
                 COEF1 = COEF(T,S,1)
                 COEF2 = COEF(T,S,2)
                 COEF3 = COEF(T,S,3)
                 COEF4 = COEF(T,S,4)
                 COEF5 = COEF(T,S,5)
C
C     COMPUTE THE COEFFICIENTS OF THE UNKOWNS IN EQUATION (I)
C
C     TEMPORARY VARIABLES
C
            DUMMY1 = COEF1*DDBX1 + COEF3*DBX1 + COEF5*BX1
            DUMMY2 = COEF1*DDBX2 + COEF3*DBX2 + COEF5*BX2
            DUMMY3 = COEF1*DDBX3 + COEF3*DBX3 + COEF5*BX3
            DUMMY4 = COEF1*DDBX4 + COEF3*DBX4 + COEF5*BX4
            DUMY1 =  COEF2*DDBY1 + COEF4*DBY1
            DUMY2  = COEF2*DDBY2 + COEF4*DBY2
            DUMY3  = COEF2*DDBY3 + COEF4*DBY3
            DUMY4  = COEF2*DDBY4 + COEF4*DBY4
C
C     COEFFICIENTS
C
            A(NROW,I1) = BY1*DUMMY1 + BX1*DUMY1
C
            A(NROW,I2) = BY3*DUMMY1 + BX1*DUMY3
C
            A(NROW,I3) = BY1*DUMMY3 + BX3*DUMY1
C
            A(NROW,I4) = BY3*DUMMY3 + BX3*DUMY3
C
            A(NROW,J1) = BY1*DUMMY2 + BX2*DUMY1
C
            A(NROW,J2) = BY3*DUMMY2 + BX2*DUMY3
C
            A(NROW,J3) = BY1*DUMMY4 + BX4*DUMY1
C
            A(NROW,J4) = BY3*DUMMY4 + BX4*DUMY3
C
            A(NROW,K1) = BY2*DUMMY2 + BX2*DUMY2
C
            A(NROW,K2) = BY4*DUMMY2 + BX2*DUMY4
C
            A(NROW,K3) = BY2*DUMMY4 + BX4*DUMY2
C
            A(NROW,K4) = BY4*DUMMY4 + BX4*DUMY4
C
            A(NROW,LI) = BY2*DUMMY1 + BX1*DUMY2
C
            A(NROW,L2) = BY4*DUMMY1 + BX1*DUMY4
C
            A(NROW,L3) = BY2*DUMMY3 + BX3*DUMY2
C
            A(NROW,L4) = BY4*DUMMY3 + BX3*DUMY4
C
C     COMPUTE THE RIGHT HAND SIDE
C
            B(NROW) = F(T,S,1)
                                                 GO TO 140
 170     CONTINUE
```

```
C     COMPUTE THE COEFFICIENTS OF THE UNKNOWNS IN THE EQUATION (II)
C
                BCOEF1 = BCOEF(T,S,1)
                BCOEF2 = BCOEF(T,S,2)
                BCOEF3 = BCOEF(T,S,3)
C
C     COMPUTE THE COEFFICIENTS OF THE UNKNOWNS IN EQUATIONS (II)
C
C     TEMPORARY VARIABLES
                BDUMY1 = BCOEF1*BY1 + BCOEF3*DBY1
                BDUMY2 = BCOEF1*BY2 + BCOEF3*DBY2
                BDUMY3 = BCOEF1*BY3 + BCOEF3*DBY3
                BDUMY4 = BCOEF1*BY4 + BCOEF3*DBY4
C
C     COEFFICIENTS
                A(NROW,I1) = BX1*BDUMY1 + BCOEF2*DBX1*BY1
                A(NROW,I2) = BX1*BDUMY3 + BCOEF2*DBX1*BY3
                A(NROW,I3) = BX3*BDUMY1 + BCOEF2*DBX3*BY1
C
                A(NROW,I4) = BX3*BDUMY3 + BCOEF2*DBX3*BY3
C
                A(NROW,J1) = BX2*BDUMY1 + BCOEF2*DBX2*BY1
C
                A(NROW,J2) = BX2*BDUMY3 + BCOEF2*DBX2*BY3
C
                A(NROW,J3) = BX4*BDUMY1 + BCOEF2*DBX4*BY1
C
                A(NROW,J4) = BX4*BDUMY3 + BCOEF2*DBX4*BY3
C
                A(NROW,K1) = BX2*BDUMY2 + BCOEF2*DBX2*BY2
C
                A(NROW,K2) = BX2*BDUMY4 + BCOEF2*DBX2*BY4
C
                A(NROW,K3) = BX4*BDUMY2 + BCOEF2*DBX4*BY2
C
                A(NROW,K4) = BX4*BDUMY4 + BCOEF2*DBX4*BY4
C
                A(NROW,L1) = BX1*BDUMY2 + BCOEF2*DBX1*BY2
C
                A(NROW,L2) = BX1*BDUMY4 + BCOEF2*DBX1*BY4
C
                A(NROW,L3) = BX3*BDUMY2 + BCOEF2*DBX3*BY2
C
                A(NROW,L4) = BX3*BDUMY4 + BCOEF2*DBX3*BY4
C
C     COMPUTE THE RIGHT HAND SIDE FROM BOUNDARY CONDITIONS.
C
                B(NROW) = F(T,S,2)
C
 140  CONTINUE
C
 110  CONTINUE
C
C     MEASURE THE EXECUTION TIME TO GENERATE MATRIX A
C     AND VECTOR B.
C
      CALL SECOND(T1)
      EXTIME = T1 - T0
      WRITE(6,200) EXTIME
 200  FORMAT(1X,34HTOTAL COEFFICIENT MATRIX FORMATION,6X,1H=,F6.3)
                                                      RETURN
      END
      SUBROUTINE BDWDTH(NWDTH)
C
C     THIS ROUTINE CALCULATES THE BANDWIDTH
C     OF THE MATRIX A
C
      DIMENSION L(2)
      COMMON/BNDCOL/NBDCO(100),E(2000)
      COMMON/ELINCD/LOLFT(100),LORGT(100),UPRGT(100),
     *              UPLFT(100),NUMEL
      INTEGER UPRGT,UPLFT
```

```
C     INITIALIZE THE NOUN LIST
      NWDTH=0
      NROW=0
C
C     VERB LIST
C
      DO 100 NEL=1,NUMEL
C     FIND THE POSITION OF NONZERO COEFFICIENT
C     FOR THE EQUAT ASSOCIATED WITH EACH
C     ELEMENT
         L(1)=4*LOLFT(NEL)-3
         L(2) = 4*UPRGT(NEL)
C
C     COMPUTES THE BANDWIDTH
         NCOLP=4+NBOCO(NEL)
            DO 102 K=1,NCOLP
C***           COMPUTES THE ROW INDEX
               NROW=NROW+1
               DO 102 N=1,2
                  NDIST=IABS(NROW-L(N))+1
                  IF(NWDTH.GT.NDIST)GOTO102
                     NWDTH=NDIST
 102           CONTINUE
 100  CONTINUE
      RETURN
      END
       FUNCTION COEF(X,Y,J)
       Z = EXP(X*Y)
       RZ = 1./Z
       GO TO (101,102,103,104,105),J
 101   COEF = Z
       RETURN
 102   COEF = RZ
       RETURN
 103   COEF = Y * [illegible]
       RETURN
 104   COEF =  X * RZ
       RETURN
 105   COEF = -1./(1. + X + Y)
       RETURN
       END
       FUNCTION F(X,Y,J)
       GO TO (101,102) , J
 101   PI = 3.14159265358979
       Z = EXP(X * Y)
       RZ = 1. / Z
       PIX = PI * X
       PIY = PI * Y
       PIZ = PI * Z
       SINX = SIN(PIX)
       SINY = SIN(PIY)
       TRUE = Z * SINX *SINY
       TEMP = PI * PI * TRUE
       XTRUE = X * TRUE
       YTRUE = Y * TRUE
       FX = PIZ * COS(PIX) *SINY
       FY = PIZ * COS(PIY) * SINX
       DXIR = YTRUE + FX
       DYIR = XTRUE + FY
       DXDX   Y*YTRUE - TEMP + 2.*Y*FX
       DYDY   X*XTRUE - TEMP + 2.*X*FY
      [illegible] TRUE/(1.+X+Y)
       RETURN
 102   F [illegible]
       RETURN
       END
       FUNCTION TRUE(X,Y)
       PI   [illegible]
       TRUE = EXP(X*Y)*SIN(PI*X)*SIN(PI*Y)
       RETURN
       END
       FUNCTION BCOEF(X,Y,J)
```