Department of Computer Science Technical Reports

Department of Computer Science

1977

ELLPACK Distribution Guide

Ronald F. Boisvert

Report Number:

77-254

ELLPACK DISTRIBUTION GUIDE

Ronald F. Boisvert
Purdue University

CSD-TR 254
December 7, 1977

(Revised September 15, 1978)

ABSTRACT

This note describes general installation procedures for the ELLPACK 77 system. This set of programs is the software product of the ELLPACK project [2] to develop a research and educational tool to evaluate software for solving elliptic partial differential equations.

SYSTEM OVERVIEW

ELLPACK 77 allows its users to pose a linear elliptic partial differential equation with general boundary conditions on a rectangular domain in two or three dimensions and to specify the names of procedures to be used to solve the problem. This ELLPACK input is coded in a simple, high-level problem-oriented language which is recognized by the ELLPACK Preprocessor. This preprocessor, a FORTRAN program, accepts this input and generates another FORTRAN program, the ELLPACK control program, which performs the requested tasks by specifying calls to FORTRAN subprograms in a library of programs called the ELLPACK modules. When the generated control program executes, then, it solves the problem specified in the original input using the methods described and produces the output requested.

Information on preparing input for ELLPACK 77 is found in [3] and details on adding modules is contained in [4]. It is assumed that the reader is familiar with both these documents in what follows.

The system described here was developed on the CDC6500 computing system at Purdue University with modules contributed from a number of sources. Various versions of ELLPACK have since run on a number of different computing systems including: IBM 370/168 (MIT, via Harvard), PDP 10 (Harvard), CDC 3170 (University of Waterloo), IBM 360/75 (University of Waterloo), and CDC 6600 (University of Texas at Austin).

The remainder of this note supplies a machine-independent description of details needed to install ELLPACK 77. In addition, a very detailed example of an ELLPACK run which illustrates the required control cards and the output generated in the Purdue implementation is given in Appendix I. In Appendix II a less detailed example of the control cards needed to run ELLPACK on an IBM/360 (OS) system is presented.

INSTALLING ELLPACK

The following facilities are required for the installation of ELLPACK: an ANSI FORTRAN compiler, a linkage-editor or linking-loader with facilities for the creation and maintenance of relocatable object program libraries, utilities for file manipulation (e.g. rewind, concatenation), and a graphics output device. The need for the latter can be eliminated by some simple changes to ELLPACK to be described later.

The first task to be performed in the installation procedure is the compilation of the Preprocessor. (Caution: read this entire document before proceeding -- some Preprocessor and ELLPACK module code may need to be adjusted for your system as described later). The compiler output (relocatable object code) should be permanently saved, say on the file PREOBJ.

Next, all ELLPACK modules should be processed by the FORTRAN compiler. However, these object modules should be permanently saved in a relocatable object program library compatible with the system linkage-editor or linking loader. Call this library ELMLIB.

We are now ready to make an ELLPACK run. As outlined previously, running ELLPACK requires two steps: running the Preprocessor and running the generated control program. The Preprocessor requires the set of files given in Table 1 (procedures for modifying the unit numbers are given later). All files except

FORTRAN logical unit number	File name	Type	Purpose
1	HEADER	Output	Control program, part 1
2	ASSIGN	Output	Control program, part 3
3	MODSEQ	Output	Control program, part 4
4	FORT	Output	Control program, part 5
5	INPUT	Input	ELLPACK input
6	OUTPUT	Output	Standard print output
7	PUNCH	Output	Currently not used
8	DATA	Output	Control program, part 2

Table 1 : Preprocessor Files

the standard printed output file and the file SCRTCH should be in card image format. Unformatted FORTRAN writes are made to the file SCRTCH. After the Preprocessor runs, the ELLPACK control program is obtained by concatenating the files HEADER, DATA, ASSIGN, MODSEQ and FORT in that order. The resulting program should then be compiled and executed, with external references resolved from ELMLIB, the relocatable object library of ELLPACK modules. The ELLPACK control program requires the set of files given in Table 2. Note that the output file SAVE contains information produced by the SAVE verb (see [3]). These include various parameters and statistics of the run which may be subsequently appended to a permanent file containing a log of ELLPACK runs that have been made. If this facility is not required the file SAVE may be either printed or discarded at the end of the run. Finally, the control program may also produce a file of commands to be executed by an external graphics device. The entire process of running an ELLPACK program is diagrammed in Figure 1.

FORTRAN logical unit number	File name	Type	Purpose
4	SAVE	Output	SAVE-ed information
5	INPUT	Input	Input to ELLPACK modules when required
6	OUTPUT	Output	Standard printed output
8	SCRTCH	Input/Output	Scratch file for modules

Table 2: Control Program Files

SYSTEM-DEPENDANT FEATURES IN ELLPACK 77

In order to increase the portability of ELLPACK, routines have been coded in ANSI FORTRAN wherever possible. In addition, the Preprocessor has been processed by the PFORT verifier. However, some facilities offered by ELLPACK

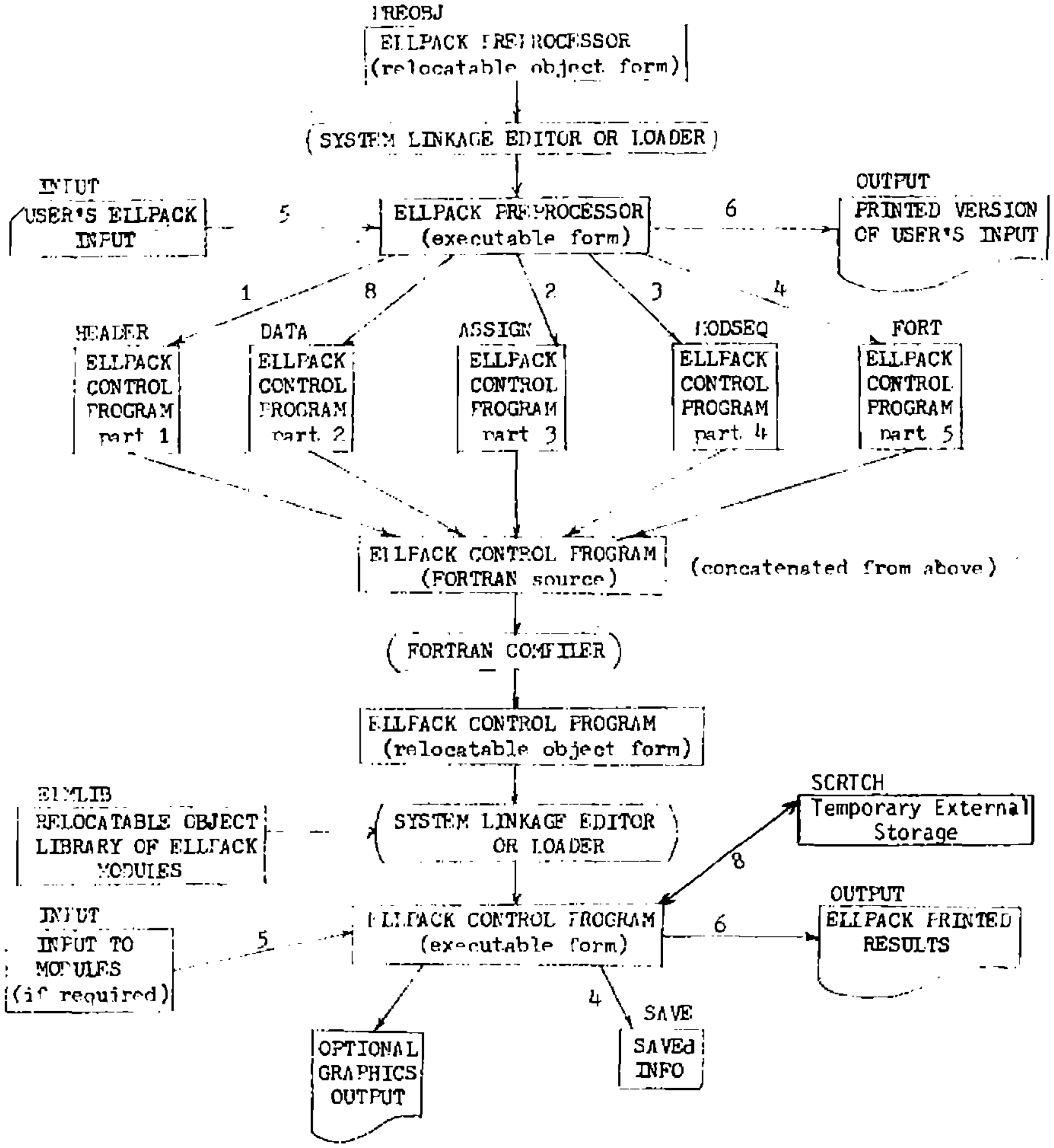

PREOBJ
ELLPACK PREPROCESSOR
(relocatable object form)
SYSTEM LINKAGE EDITOR OR LOADER
INPUT
USER'S ELLPACK INPUT
5
ELLPACK PREPROCESSOR
(executable form)
6
OUTPUT
PRINTED VERSION OF USER'S INPUT
1
8
2
3
4
HEADER
DATA
ASSIGN
MODSEQ
FORT
ELLPACK CONTROL PROGRAM part 1
ELLPACK CONTROL PROGRAM part 2
ELLPACK CONTROL PROGRAM part 3
ELLPACK CONTROL PROGRAM part 4
ELLPACK CONTROL PROGRAM part 5
ELLPACK CONTROL PROGRAM
(FORTRAN source)
(concatenated from above)
FORTRAN COMPILER
ELLPACK CONTROL PROGRAM
(relocatable object form)
ELMLIB
RELOCATABLE OBJECT LIBRARY OF ELLPACK MODULES
SYSTEM LINKAGE EDITOR OR LOADER
SCRTCH
Temporary External Storage
8
INPUT
INPUT TO MODULES (if required)
5
ELLPACK CONTROL PROGRAM
(executable form)
6
OUTPUT
ELLPACK PRINTED RESULTS
4
SAVE
SAVEd INFO
OPTIONAL GRAPHICS OUTPUT

Figure 1 : Running an ELLPACK Program

Names above boxes are file names and numbers above arrows are FORTRAN logical unit numbers.

are inherently system dependent and thus it may be necessary to make a number of changes or additions to the source code before it is compiled. The following items should be scrutinized.

1. Required utilities. The ELLPACK Preprocessor and the generated control program each use the subroutine subprogram SECOND with one real parameter T. When invoked, SECOND sets T to the elapsed CP time in seconds since the start of this job. This routine is called whenever TIME is specified in the OPTIONS segment of the ELLPACK input. If a similar routine is not available, then a dummy routine returning 0 should be placed in the libraries PREOBJ and ELMLIB. Also required is a real function subprogram DATE with one (unused) integer parameter. DATE returns the current date in A8 format (MM/DD/YY). A call to this routine is generated when SAVE-PARAMETERS is specified in the OUTPUT segment of the ELLPACK input. On systems with small word sizes it may be necessary to eliminate this routine or make other adjustments. This can be done by modifying FORMAT statements 670 and 671 in the Preprocessor routine OUTPT.

2. Program cards. The main programs of the Preprocessor and of the generated ELLPACK control program each begin with the non-standard PROGRAM statement which names the programs and equivalences unit numbers with external file names. These may be removed by deleting the appropriate lines in the Preprocessor main program and by modifying FORMAT statement number 601 in the Preprocessor routine HEADIT.

3. Logical unit numbers in the Preprocessor. If it is necessary to change the logical unit numbers for the Preprocessor files described in Table 1, the assignments made to the variables MHEAD, MASSGN, MMODSQ, MFORT, MINPUT, MOUTPT, MPUNCH and MDATA in a data statement in the Preprocessor BLOCK DATA subprogram should be modified.

4. Logical unit numbers in the ELLPACK control program. If it is necessary to change the unit numbers 5 and 6 in use by the control program, one must modify the assignments to the variables MINPUT and MOUTPT generated by the Preprocessor routine HEADIT via FORMAT statement 604. If unit number 4 must also be changed, the write statements generated via FORMAT statements 620 and 670 in the Preprocessor routine OUTPT must be modified.

5. Plotting. The current version of ELLPACK has facilities to produce domain plots and contour plots (of error, true solution, residual or computed solution). Routines to perform these tasks are implemented using the "standard" FORTRAN callable Calcomp plotting package [1]. Plots at Purdue are produced on a Versatec electrostatic printer. It may be necessary to modify the ELLPACK output modules regardless of whether or not plotting is to be included in the system. The affected ELLPACK modules and their uses are:

REGPLT - Called when PLOT-DOMAIN is specified. Plots boundary of rectangular region, grid lines and axes. A rewrite may be necessary.

DOMPLT - Called when PLOT-DOMAIN is specified and the domain is non-rectangular. Plots boundary of non-rectangular region, grid lines and axes. A rewrite may be necessary.

CONTUR - Called when PLOT-TRUE, PLOT-U or PLOT-ERROR are specified. Sets up the interface for the contour plotter GCONTR and labels plots. A rewrite may be necessary.

GCONTR - A portable device-independent contour plotting program [5]. No modifications are necessary.

DRAW - Does actual plotting for GCONTR. A rewrite may be necessary. See the machine-readable documentation in the routine GCONTR.

IGET, MARK1, FILLO - GCONTR utilities. May be necessary to modify the data statement assigning a value to NBPW (The number of bits per word used in integer arithmetic).

Finally, the initialization and finalization calls to the plotting system generated via FORMAT statements 615 and 616 in the Preprocessor routine CLOSER must also be modified or deleted.

To remove plotting from ELLPACK, simply remove the ELLPACK routines described above. In this way, the specifications PLOT-DOMAIN, PLOT-ERROR, PLOT-TRUE, PLOT-RESIDUAL and PLOT-U will still be recognized by the Preprocessor, but unsatisfied external references will occur when they are specified.

6. Machine-dependent constants. One machine-dependent constant is currently written onto the ELLPACK control program by the Preprocessor. This constant, EPSGRD, is the tolerance for distinguishing points from the grid. Its value is set at 10^{-8} for the Purdue CDC system (14.3 digits). To change this value, FORMAT statement number 608 in the Preprocessor routine HEADIT must be modified.

7. System dependencies in the ELLPACK modules. In this section we have attempted to outline possible sources of difficulty in installing the ELLPACK Preprocessor and the Purdue output routines. No attempt has been made to describe non-standard features contained in the ELLPACK modules themselves. ELLPACK contributors have been urged to use ANS standard FORTRAN and to avoid the use of system-dependent features. We will be happy to forward news of any difficulties experienced to the appropriate contributor.

REFERENCES

[1] Dodson, D. S., "User's Guide to Plotting," Purdue Univ. Comp. Ctr. Report J5 CALCOMP, Aug. 1977.

[2] Rice, J. R., "ELLPACK Cooperative Group Formation," SIGNUM Newsletter, Vol. 11, No. 3, Oct. 1976, p. 16.

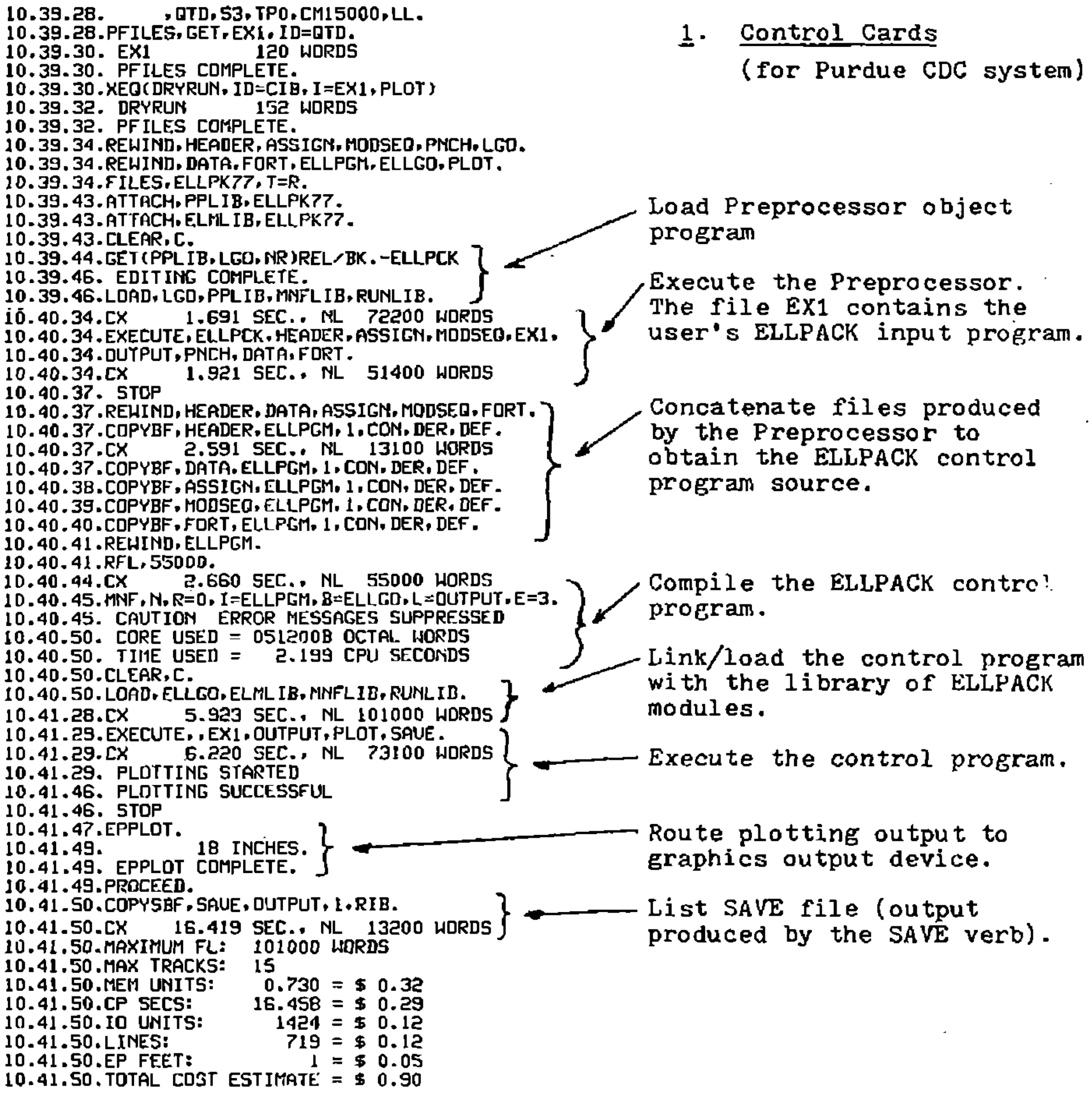
[3] Rice, J. R., "ELLPACK 77 User's Guide," Purdue Univ. Comp. Sci. Dept. Report CSD-TR 226, Mar. 18, 1977 (Revised, Sept. 1978).

[4] Rice, J. R., "ELLPACK 77 Contributor's Guide," Purdue Univ. Comp. Sci. Dept. Report CSD-TR 267, June 10, 1978.

[5] Snyder, W. V., "Contour Plotting," Jet Propulsion Lab, Section 366, Computing Memo, No. 433, Nov. 8, 1977.

APPENDIX 1

The following sample ELLPACK 77 run illustrates the Purdue implementation of the ELLPACK system.



```
10.39.28.      ,QTD,S3,TPO,CM15000,LL.
10.39.28.PFILES,GET,EX1,ID=QTD.
10.39.30. EX1           120 WORDS
10.39.30. PFILES COMPLETE.
10.39.30.XEQ(DRYRUN,ID=CIB,I=EX1,PLOT)
10.39.32. DRYRUN        152 WORDS
10.39.32. PFILES COMPLETE.
10.39.34.REWIND,HEADER,ASSIGN,MODSEQ,PNCH,LGO.
10.39.34.REWIND,DATA,FORT,ELLPGM,ELLGO,PLOT.
10.39.34.FILES,ELLPK77,T=R.
10.39.43.ATTACH,PPLIB,ELLPK77.
10.39.43.ATTACH,ELMLIB,ELLPK77.
10.39.43.CLEAR,C.
10.39.44.GET(PPLIB,LGO,NR)REL/BK.-ELLPCK
10.39.46. EDITING COMPLETE.
10.39.46.LOAD,LGO,PPLIB,MNFLIB,RUNLIB.
10.40.34.CX      1.691 SEC., NL  72200 WORDS
10.40.34.EXECUTE,ELLPCK,HEADER,ASSIGN,MODSEQ,EX1.
10.40.34.OUTPUT,PNCH,DATA,FORT.
10.40.34.CX      1.921 SEC., NL  51400 WORDS
10.40.37. STOP
10.40.37.REWIND,HEADER,DATA,ASSIGN,MODSEQ,FORT.
10.40.37.COPYBF,HEADER,ELLPGM,1,CON,DER,DEF.
10.40.37.CX      2.591 SEC., NL  13100 WORDS
10.40.37.COPYBF,DATA,ELLPGM,1,CON,DER,DEF.
10.40.38.COPYBF,ASSIGN,ELLPGM,1,CON,DER,DEF.
10.40.39.COPYBF,MODSEQ,ELLPGM,1,CON,DER,DEF.
10.40.40.COPYBF,FORT,ELLPGM,1,CON,DER,DEF.
10.40.41.REWIND,ELLPGM.
10.40.41.RFL,55000.
10.40.44.CX      2.660 SEC., NL  55000 WORDS
10.40.45.MNF,N,R=0,I=ELLPGM,B=ELLGO,L=OUTPUT,E=3.
10.40.45. CAUTION  ERROR MESSAGES SUPPRESSED
10.40.50. CORE USED = 051200B OCTAL WORDS
10.40.50. TIME USED =   2.199 CPU SECONDS
10.40.50.CLEAR,C.
10.40.50.LOAD,ELLGO,ELMLIB,MNFLIB,RUNLIB.
10.41.28.CX      5.923 SEC., NL 101000 WORDS
10.41.29.EXECUTE,,EX1,OUTPUT,PLOT,SAVE.
10.41.29.CX      6.220 SEC., NL  73100 WORDS
10.41.29. PLOTTING STARTED
10.41.46. PLOTTING SUCCESSFUL
10.41.46. STOP
10.41.47.EPPLOT.
10.41.49.            18 INCHES.
10.41.49. EPPLOT COMPLETE.
10.41.49.PROCEED.
10.41.50.COPYSBF,SAVE,OUTPUT,1,RIB.
10.41.50.CX     16.419 SEC., NL  13200 WORDS
10.41.50.MAXIMUM FL:  101000 WORDS
10.41.50.MAX TRACKS:  15
10.41.50.MEM UNITS:     0.730 = $ 0.32
10.41.50.CP SECS:      16.458 = $ 0.29
10.41.50.IO UNITS:       1424 = $ 0.12
10.41.50.LINES:           719 = $ 0.12
10.41.50.EP FEET:           1 = $ 0.05
10.41.50.TOTAL COST ESTIMATE = $ 0.90
```

1. Control Cards
(for Purdue CDC system)

Load Preprocessor object program

Execute the Preprocessor. The file EX1 contains the user's ELLPACK input program.

Concatenate files produced by the Preprocessor to obtain the ELLPACK control program source.

Compile the ELLPACK control program.

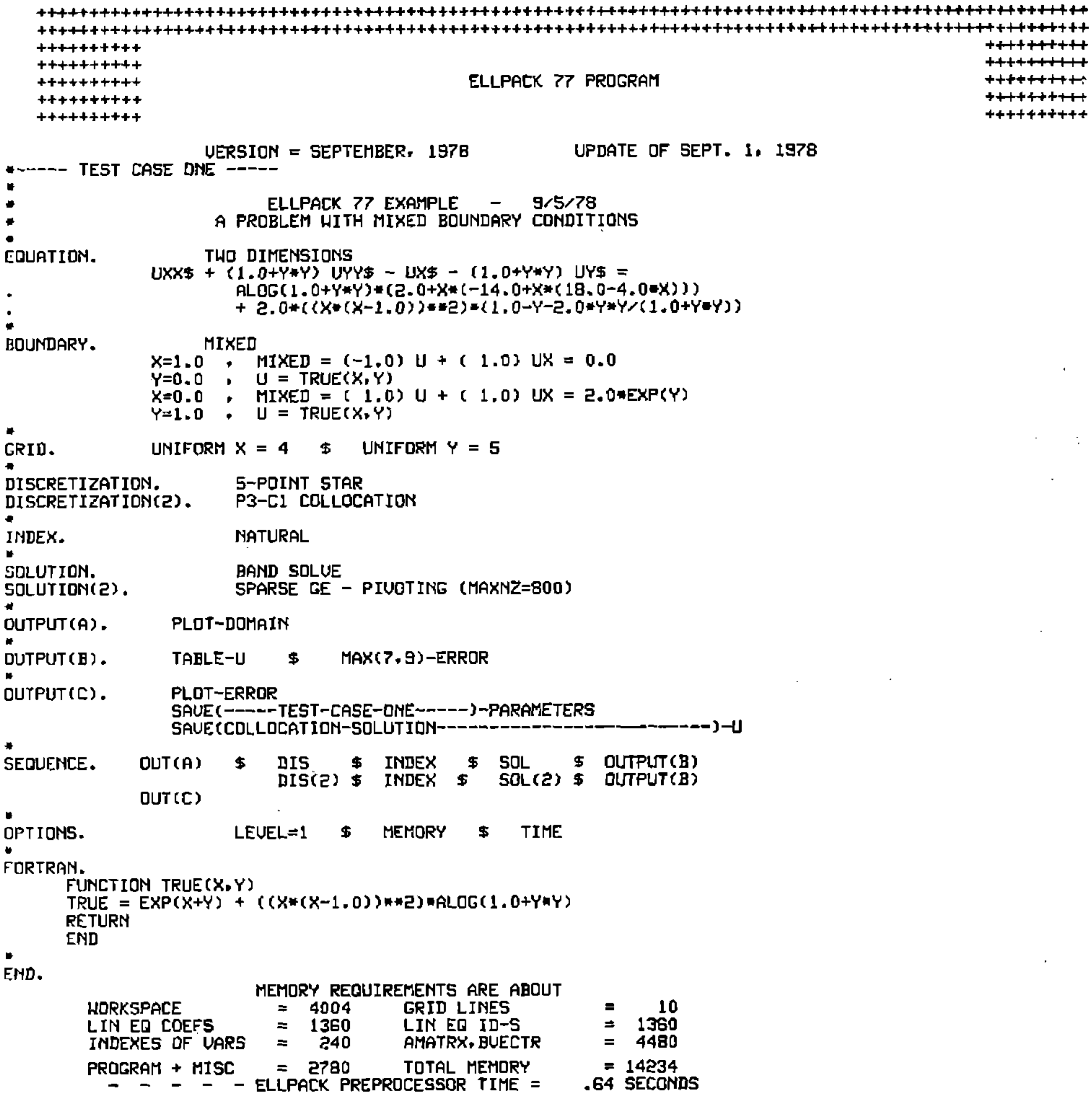
Link/load the control program with the library of ELLPACK modules.

Execute the control program.

Route plotting output to graphics output device.

List SAVE file (output produced by the SAVE verb).

2. The ELLPACK Input Program (read by the Preprocessor)



```
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
   ++++++++++                                                                                ++++++++++
   ++++++++++                                                                                ++++++++++
   ++++++++++                                ELLPACK 77 PROGRAM                              ++++++++++
   ++++++++++                                                                                ++++++++++
   ++++++++++                                                                                ++++++++++

                   VERSION = SEPTEMBER, 1978          UPDATE OF SEPT. 1, 1978
*----- TEST CASE ONE -----
*
*                        ELLPACK 77 EXAMPLE    -    9/5/78
*                   A PROBLEM WITH MIXED BOUNDARY CONDITIONS
*
EQUATION.          TWO DIMENSIONS
              UXX$ + (1.0+Y*Y) UYY$ - UX$ - (1.0+Y*Y) UY$ =
.                     ALOG(1.0+Y*Y)*(2.0+X*(-14.0+X*(18.0-4.0*X)))
.                     + 2.0*((X*(X-1.0))**2)*(1.0-Y-2.0*Y*Y/(1.0+Y*Y))
*
BOUNDARY.          MIXED
              X=1.0  ,  MIXED = (-1.0) U + ( 1.0) UX = 0.0
              Y=0.0  ,  U = TRUE(X,Y)
              X=0.0  ,  MIXED = ( 1.0) U + ( 1.0) UX = 2.0*EXP(Y)
              Y=1.0  ,  U = TRUE(X,Y)
*
GRID.         UNIFORM X = 4   $   UNIFORM Y = 5
*
DISCRETIZATION.       5-POINT STAR
DISCRETIZATION(2).    P3-C1 COLLOCATION
*
INDEX.                NATURAL
*
SOLUTION.             BAND SOLVE
SOLUTION(2).          SPARSE GE - PIVOTING (MAXNZ=800)
*
OUTPUT(A).       PLOT-DOMAIN
*
OUTPUT(B).       TABLE-U    $    MAX(7,9)-ERROR
*
OUTPUT(C).       PLOT-ERROR
                 SAVE(-----TEST-CASE-ONE-----)-PARAMETERS
                 SAVE(COLLOCATION-SOLUTION---------------------------)-U
*
SEQUENCE.    OUT(A)   $   DIS    $  INDEX   $  SOL    $  OUTPUT(B)
                          DIS(2) $  INDEX  $   SOL(2) $  OUTPUT(B)
             OUT(C)
*
OPTIONS.              LEVEL=1   $   MEMORY   $   TIME
*
FORTRAN.
      FUNCTION TRUE(X,Y)
      TRUE = EXP(X+Y) + ((X*(X-1.0))**2)*ALOG(1.0+Y*Y)
      RETURN
      END
*
END.
                        MEMORY REQUIREMENTS ARE ABOUT
        WORKSPACE          =  4004     GRID LINES          =     10
        LIN EQ COEFS       =  1360     LIN EQ ID-S         =   1360
        INDEXES OF VARS    =   240     AMATRX,BVECTR       =   4480

        PROGRAM + MISC     =  2780     TOTAL MEMORY        =  14234
          -  -  -  -  - ELLPACK PREPROCESSOR TIME =     .64 SECONDS
```

3. The ELLPACK Control Program (generated by the Preprocessor)

```
MINNESOTA FORTRAN COMPILER (VERSION 5.2 - 02/23/78) ON PURDUE DUAL 6500 MACE SYSTEM  09/14/78 AT 10.43

    MNF,N,R=0,I=ELLPGM,B=ELLGO,L=OUTPUT,E=3.

    C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    C     ++++++++++                                        ++++++++++
    C     ++++++++++        FILE HEADER   FOR ELLPACK 77    ++++++++++
    C     ++++++++++                                        ++++++++++
           PROGRAM ELL77 (INPUT,OUTPUT,PLOT,SAVE,SCRTCH,TAPE4=SAVE.
          A              TAPE5=INPUT,TAPE6=OUTPUT,TAPE8=SCRTCH)
           LOGICAL DIM2,DIM3,POISON.LAPLAC,CONSTC,SELFAD,CROSST.
          A        DIRICH,HOMOEQ.NEUMAN,MIXED,HOMOBC,UNIFRM.
          B        DEBUG,TIMER.RECTAN,SYMMET,CLOCKW,FATAL
           REAL AX,BX,AY,BY.AZ,BZ,HX,HY,HZ,CUXX,CUXY,CUYY.CUX.
          A     CUY,CU,CUXZ,CUYZ,CUZZ,CUZ.GRIDX,GRIDY,GRIDZ.
          B     AMATRX.BVECTR,UNKNWN.COEF,TABLEM.BPARAM,XBOUND,
          C     YBOUND
           INTEGER IROT(6),HORZ.VERT,BOTH,STAR.ONE,BLANK,XXX,EXTER.
          A        PERIOD,CORNER
           EXTERNAL SOLUT,ERROR,RESID,TRUEEP
           COMMON WORKSP
           COMMON/ PROBL / DIM2,DIM3,POISON.LAPLAC,CONSTC,SELFAD.
          A        HOMOEQ,CROSST,HOMOBC.DIRICH.NEUMAN,MIXED,UNIFRM
           COMMON/ PROBR / AX,BX,AY,BY,AZ,BZ,HX,HY,HZ
           COMMON / PROBI /NGRIDX,NGRIDY,NGRIDZ
           COMMON/ CPDE  / CUXX,CUXY.CUYY.CUX.CUY,CU,
          A                CUXZ,CUYZ.CUZZ.CUZ
           COMMON/ CONTRL/ DEBUG,TIMER,SYMMET.FATAL
           COMMON/ INTEGS/ NUMBEQ,NUMCOE.NROW.NCOL,NBAND,IROT,
          A                LEVEL,INITL.INDIS,INSOL.
          B                MINPUT,MOUTPT.MEMORY,ININD.NGRDXD.
          C                NGRDYD,NGRDZD,MXNEQ,MXNCOE,NROWD,NCOLD
           COMMON / REALS / ERRMAX.SOLMAX,TRUMAX,RESMAX,TIMES
           COMMON/ BNDRY / IPIECE.NBOUND,NBNDPT,CLOCKW
           COMMON/SYMCON / HORZ,VERT,BOTH,INTER,STAR,ONE.BLANK,MINUS,
          A                 XXX,EXTER,PERIOD,CORNER
           COMMON/ NUMCON / EPSGRD
           COMMON / COEFZZ / COEF
           COMMON / IDCOZZ / IDCOEF
           COMMON / NDXEZZ / NDXEQ
           COMMON / NDXUZZ / NDXUNK
           COMMON / INUNZZ / INUNDX
           COMMON / AMATZZ / AMATRX
           COMMON / BVECZZ / BVECTR
           COMMON / UNKNZZ / UNKNWN
           COMMON / GRIDXZ / GRIDX
           COMMON / GRIDYZ / GRIDY
           COMMON / GRIDZZ / GRIDZ
           COMMON / TABLZZ / TABLEM
           COMMON / IGRIZZ / IGRID
           COMMON / BRANZZ / BRANGE
           COMMON / BCTYZZ / BCTYPE
           COMMON / GTYPZZ / GTYPE
           COMMON / XBOUZZ / XBOUND
           COMMON / YBOUZZ / YBOUND
           COMMON / PIECZZ / PIECE
           COMMON / BPTYZZ / BPTYPE
           COMMON / BNEIZZ / BNEIGH
           COMMON / BPARZZ / BPARAM
           COMMON / BGRIZZ / BGRID
           DIMENSION GRIDX(4      )
           DIMENSION GRIDY(5      )
           DIMENSION GRIDZ(1      )
           DIMENSION MODSQ(11)
           DIMENSION WORKSP(  4004)
```

Program card for CDC systems

```
      DIMENSION COEF(   80,17)
      DIMENSION IDCOEF(  80,17)
      DIMENSION NDXEQ(  80),NDXUNK(  80),INVNDX(  80),IGRID(  80)
      DIMENSION AMATRX(  80,  55),UNKNWN(  80),BVECTR(  80)
      INTEGER BCTYPE(    6)
      INTEGER   GTYPE(  1,  1)
      REAL      BRANGE(2,  1)
      DIMENSION XBOUND(  1),YBOUND(  1),BPARAM(  1)
      INTEGER   PIECE (  1),BPTYPE(  1),BNEIGH(  1)
      INTEGER   BGRID (  1)
      DIMENSION TABX(    7),TABY(    9),TABZ(    1)
      DIMENSION TABLEM(    4,    5,    1)
      INTEGER NAMES(4,2)
      REAL               TIMES(  14)
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++                                        ++++++++++
C    ++++++++++      FILE DATA     FOR ELLPACK 77      ++++++++++
C    ++++++++++                                        ++++++++++
      DATA  HORZ    ,    VERT  ,  BOTH   ,  INTER  ,  STAR   ,   ONE
     A  /4HHORZ    ,4HVERT    , 4HBOTH  ,5HINTER  , 4H****  ,  1H1    /
      DATA BLANK    ,  MINUS  ,    XXX   ,  EXTER  ,  PERIOD , CORNER
     A  / 4H       , 4H----  , 4HXXXX  , 5HEXTER , 1H.      , 6HCORNER /
      DATA EPSGRD / .00000001 /
      DATA MODSQ / 6, 1, 3, 4, 7, 2, 3, 5, 7, 8,
     B              9/
      DATA NAMES(1,1),NAMES(1,2),NAMES(2,1),NAMES(2,2),NAMES(3,1),
     A      NAMES(3,2),NAMES(4,1),NAMES(4,2)
     B      / 4HRESI,4HDUAL,4HSOLU,4HTION,4HTRUE,4H    ,4HERRO,4HR      /
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++                                        ++++++++++
C    ++++++++++      FILE ASSIGN   FOR ELLPACK 77      ++++++++++
C    ++++++++++                                        ++++++++++
      DIM2   = .TRUE.
      DIM3   = .FALSE.
      POISON = .FALSE.
      LAPLAC = .FALSE.
      CONSTC = .FALSE.
      SELFAD = .FALSE.
      HOMOEQ = .FALSE.
      HOMOBC = .FALSE.
      CROSST = .FALSE.
      RECTAN = .FALSE.
      UNIFRM = .FALSE.
      DIRICH = .TRUE.
      NEUMAN = .FALSE.
      MIXED  = .FALSE.
      DEBUG  = .FALSE.
      TIMER  = .FALSE.
      MINPUT = 5
      MOUTPT = 6
      DO 5 I = 1,6
    5     IROT(I) = 0
      IROT(1) = 2
      IROT(3) = -2
      LEVEL = 1
      MXNUNK = 1
      CLOCKW = .TRUE.
      SYMMET = .FALSE.
      AZ = 0.0
      BZ = 0.0
      ERRMAX = 0.0
      RESMAX = 0.0
      HZ = 0.0
      MEMORY = 0
      TFIRST = 0.0
      TLAST = 0.0
      KTIMES = 1
      FATAL  = .FALSE.
      MIXED = .TRUE.
      MIXED = .TRUE.
      AX = 1.0
      BX = 0.0
      IF( AX .LT. BX ) GO TO 210
```

Machine-
dependant
constant

Note: these headings distinguish
separate Preprocessor output
files that have been concatenated.

The initial ELLPACK interface
is set up here.

```
      ASAVE = AX
      AX = BX
      BX = ASAVE
      IROT(1) = 0
      IROT(3) = 0
  210 CONTINUE
      AY = 0.0
      BY = 1.0
      IF( AY .LT. BY ) GO TO 220
      ASAVE = AY
      AY = BY
      BY = ASAVE
      IROT(2) = +2
      IROT(4) = -2
  220 CONTINUE
      IR = 1 + IROT(1)
      BCTYPE(IR) = 3
      IR = 2 + IROT(2)
      BCTYPE(IR) = 1
      IR = 3 + IROT(3)
      BCTYPE(IR) = 3
      IR = 4 + IROT(4)
      BCTYPE(IR) = 1
      NGRDXD = 4
      NGRIDX = 4
      HX = (BX-AX)/  3.
      DO 11 J = 1,  3
   11     GRIDX(J) = AX + FLOAT(J-1)*HX
      GRIDX(  4) = BX
      NGRIDY = 5
      NGRDYD = 5
      HY = (BY-AY)/  4.
      DO 12 J = 1,  4
   12     GRIDY(J) = AY + FLOAT(J-1)*HY
      GRIDY(  5) = BY
      UNIFRM = .TRUE.
      NGRDZD = 1
      NGRIDZ = 1
      LEVEL = 1
      TIMER = .TRUE.
      NPDIM =    4
      NBDIM  =    1
      MXNCOE =  17
      MXNEQ  =  80
      NROWD  =  80
      NCOLD  =  55
      NBAND  =  27
      MEMORY =   14234
      DO 9 J=1,MXNEQ
      NDXEQ(J) = -10001
      NDXUNK(J) = -10001
      INVNDX(J) = -10001
    9 IGRID(J) = -10001
      DO 7 I = 1,   14
    7     TIMES(I) = 0.
      CALL SECOND(TFIRST)
      TIMES(  14) = TFIRST
      CALL PLOTS
C   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C   +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C   ++++++++++                                       ++++++++++
C   ++++++++++       FILE SEQUENCE FOR ELLPACK 77    ++++++++++
C   ++++++++++                                       ++++++++++
      KSEG = 0
                                                       GO TO 600
C
  500 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INITL = 1
      INDIS =  1
      CALL STAR5 (GRIDX,NGRDXD,GRIDY,NGRDYD,BCTYPE,COEF,MXNCOE,
     A            MXNEQ,IDCOEF,IGRID)
      IF( .NOT. TIMER )                    GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
```

Call on timing routine.

Initialization of plotting system.

DIS

```
C
  501 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INITL = 1
      INDIS =  6
      CALL P3C1CO(GRIDX,NGRDXD,GRIDY,NGRDYD,COEF,MXNCOE,MXNEQ,       DIS(2)
     A            IDCOEF,IGRID)
      IF( .NOT. TIMER )                      GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  502 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)                                 INDEX
      INITL = 1
      ININD =  3
      CALL NATORD(NDXEQ,NDXUNK,INVNDX,MXNEQ)
      IF( .NOT. TIMER )                      GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  503 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INSOL =   2                                                     SOL
      CALL BNDSTR(COEF,MXNCOE,MXNEQ,IDCOEF,AMATRX,NROWD,
     A            NCOLD,BVECTR,NDXEQ,NDXUNK,INVNDX)
      IF( TIMER ) CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST - TFIRST
      KTIMES = KTIMES + 1
      TFIRST = TLAST
      CALL BNDSOL(AMATRX,NROWD,NCOLD,UNKNWN,BVECTR,MXNEQ)
      IF( .NOT. TIMER )                      GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  504 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INSOL =   3
      MAXNZ=800                                                       SOL(2)
      CALL SPASTR(COEF,MXNEQ,MXNCOE,IDCOEF,AMATRX,NROWD,
     A            NCOLD,BVECTR,NDXUNK,INVNDX,MAXNZ)
      IF( TIMER ) CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST - TFIRST
      KTIMES = KTIMES + 1
      TFIRST = TLAST
      CALL SPAGEP(AMATRX,NROWD,NCOLD,UNKNWN,BVECTR,MXNEQ,NDXEQ,MAXNZ)
      IF( .NOT. TIMER )                      GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  505 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INITL = 1
      CALL REGPLT(GRIDX,GRIDY,NGRIDX,NGRIDY,AX,BX,AY,BY)              OUTPUT(A)
      IF( .NOT. TIMER )                      GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  506 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INITL = 1
      CALL TABLER(SOLUT ,NAMES(2,1),NAMES(2,2),
     $            NGRIDX,NGRIDY,NGRIDZ,GRIDX,GRIDY,GRIDZ,             OUTPUT(B)
     A            NGRDXD,NGRDYD,NGRDZD,GRIDX,GRIDY,GRIDZ,
     B            UNKNWN,MXNEQ,NDXUNK,TABLEM,COEF,IDCOEF,
     C            MXNCOE,BCTYPE,INVNDX,NDXEQ,IGRID)
```

```
      NTABX =      7
      NTABY =      9
      NTABZ =      1
      STEP = (BX-AX)/  6.
      DO   701 I=1,  6
  701     TABX(I) = AX + FLOAT(I-1)*STEP
      TABX(  7) = BX
      STEP = (BY-AY)/  8.
      DO   702 I=1,  8
  702     TABY(I) = AY + FLOAT(I-1)*STEP
      TABY(  9) = BY
      ERRMAX = FNCMAX(ERROR ,NAMES(4,1),NAMES(4,2),
     $            NTABX,NTABY,NTABZ,TABX,TABY,TABZ,
     A             NGRDXD,NGRDYD,NGRDZD,GRIDX,GRIDY,GRIDZ,
     B             UNKNWN,MXNEQ,NDXUNK,TABLEM,COEF,IDCOEF,
     C             MXNCOE,BCTYPE,INUNDX,NDXEQ,IGRID)
      IF( .NOT. TIMER )                        GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600
C
  507 CONTINUE
      IF( TIMER ) CALL SECOND(TFIRST)
      INITL = 1
      CALL CONTUR(ERROR ,NAMES(4,1),NAMES(4,2),
     A             NGRDXD,NGRDYD,NGRDZD,GRIDX,GRIDY,GRIDZ,
     B             UNKNWN,MXNEQ,NDXUNK,TABLEM,COEF,IDCOEF,
     C             MXNCOE,BCTYPE,INUNDX,NDXEQ,IGRID)
      WRITE(4,903)
  903 FORMAT(23H-----TEST-CASE-ONE-----
     *)
      RDATE = DATE(0)
      WRITE(4,902) RDATE,DIM2,POISON,LAPLAC,CONSTC,SELFAD,
     A       CROSST,HOMOEQ,AX,BX,AY,BY,AZ,BZ,DIRICH,NEUMAN,
     B       MIXED,HOMOBC,(BCTYPE(I),I=1,6),
     C       UNIFRM,NGRIDX,NGRIDY,NGRIDZ,HX,HY,HZ,
     D       (J,TIMES(J),J=1,KTIMES)
      WRITE(4,911) MEMORY,ERRMAX,RESMAX
  902 FORMAT(20X,7HDATE = , A8/
     A 2X,41HDIM2,POISON,LAPLAC,CONSTC,SELFAD,CROSST,
     B ,8HHOMOEQ =,7L3 / 2X,20HAX,BX,AY,BY,AZ,BZ = ,
     C 6F9.6  / 2X,28HDIRICH,NEUMAN,MIXED,HOMOBC =,4L3,
     D 11H   BCTYPE =,6I2  / 2X,8HUNIFRM =,L3,
     E 15H   NGRIDX,Y,Z =,3I3,13H   HX,HY,HZ =,3F10.7 /
     F 2X,22HTABLE OF SEGMENT TIMES / (6X,7(I3,F7.3)))
  911 FORMAT(11H   MEMORY =,I5,5X,20HMAX ERROR,RESIDUAL =,2E14.4)
      WRITE(4,904)
  904 FORMAT(40HCOLLOCATION-SOLUTION---------------------
     *, 6H------
     *)
  901 FORMAT(I10/(6E13.6))
      WRITE(4,901)NUMBEQ,(UNKNWN(J),J=1,NUMBEQ)
      IF( .NOT. TIMER )                        GO TO 600
      CALL SECOND(TLAST)
      TIMES(KTIMES) = TLAST-TFIRST
      KTIMES = KTIMES+1
      GO TO 600

  598 CONTINUE
  600 IF( FATAL )  GO TO 690
      KSEG = KSEG + 1
      IF( KSEG .GT. 10)     GO TO 605
      MSEG = MODSQ(KSEG)
      GO TO(500,501,502,503,504,505,506,507,
     B                    600,598,650), MSEG
  605 CONTINUE
  650 CONTINUE
```

OUTPUT(C)

Call to system
date routine.

```
      CALL SECOND(TFINAL)
      TIMES(KTIMES) = TFINAL - TIMES(  14)
      WRITE(MOUTPT,700) (I,TIMES(I),I=1,KTIMES)
  700 FORMAT(/////3X,34HEXECUTION TIME FOR MODULES  (SEC.)/
     A  3X,26(1H-)//(4(5X,I3,1X,6HTIME =,F7.2,3X)/))
      CALL PLOT(0.,0.,999)
      STOP
  690 CONTINUE
      STOP
      END
```

Finalization of plotting system.

```
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C    ++++++++++                                        ++++++++++
C    ++++++++++       FILE FORTRAN  FOR ELLPACK 77     ++++++++++
C    ++++++++++                                        ++++++++++

      SUBROUTINE PDE(X,Y,CVALUS)
      REAL            CVALUS(7)
      CVALUS( 1) = 1.0
      CVALUS( 2) = 0.0
      CVALUS( 3) = +(1.0+Y*Y)
      CVALUS( 4) = -1.0
      CVALUS( 5) = -(1.0+Y*Y)
      CVALUS( 6) = 0.0
      CVALUS( 7) = ALOG(1.0+Y*Y)*(2.0+X*(-14.0+X*(18.0-4.0*X)))+2.0*((X*
     *(X-1.0))**2)*(1.0-Y-2.0*Y*Y/(1.0+Y*Y))
      RETURN
      END

      REAL            FUNCTION PDERHS(X,Y)
      PDERHS = ALOG(1.0+Y*Y)*(2.0+X*(-14.0+X*(18.0-4.0*X)))+2.0*(
     +(X*(X-1.0))**2)*(1.0-Y-2.0*Y*Y/(1.0+Y*Y))
      RETURN
      END

      REAL            FUNCTION BCOND(I,X,Y,BVALUS)
      REAL            BVALUS(4)
      COMMON / INTEGS / NUMBEQ,NUMCOE,NROW,NCOL,NBAND,IROT(6),
     A LEVEL,INITL,INDIS,INSOL,MINPUT,MOUTPT,MEMORY,ININD,
     B NGRDXD,NGRDYD,NGRDZD,MXNEQ,MXNCOE,NROWD,NCOLD
      IP = I + IROT(I)
      GOTO( 1, 2, 3,
     B    4),IP
    1 CONTINUE
      BVALUS(4) =0.0
      BVALUS(1) =-1.0
      BVALUS(2) =1.0
      BVALUS(3) = 0.0
                 GO TO 999
    2 BVALUS(1) = 1.0
      BVALUS(4) =TRUE(X,Y)
      BVALUS(2) = 0.0
      BVALUS(3) = 0.0
                 GO TO 999
    3 CONTINUE
      BVALUS(4) =2.0*EXP(Y)
      BVALUS(1) =1.0
      BVALUS(2) =1.0
      BVALUS(3) = 0.0
                 GO TO 999
    4 BVALUS(1) = 1.0
      BVALUS(4) =TRUE(X,Y)
      BVALUS(2) = 0.0
      BVALUS(3) = 0.0
                 GO TO 999
  999 BCOND = BVALUS(4)
      RETURN
      END
```

User routine from FORTRAN segment.

```
      FUNCTION TRUE(X,Y)
      TRUE = EXP(X+Y) + ((X*(X-1.0))**2)*ALOG(1.0+Y*Y)
      RETURN
      END
```

4. The ELLPACK Output (produced by ELLPACK modules)

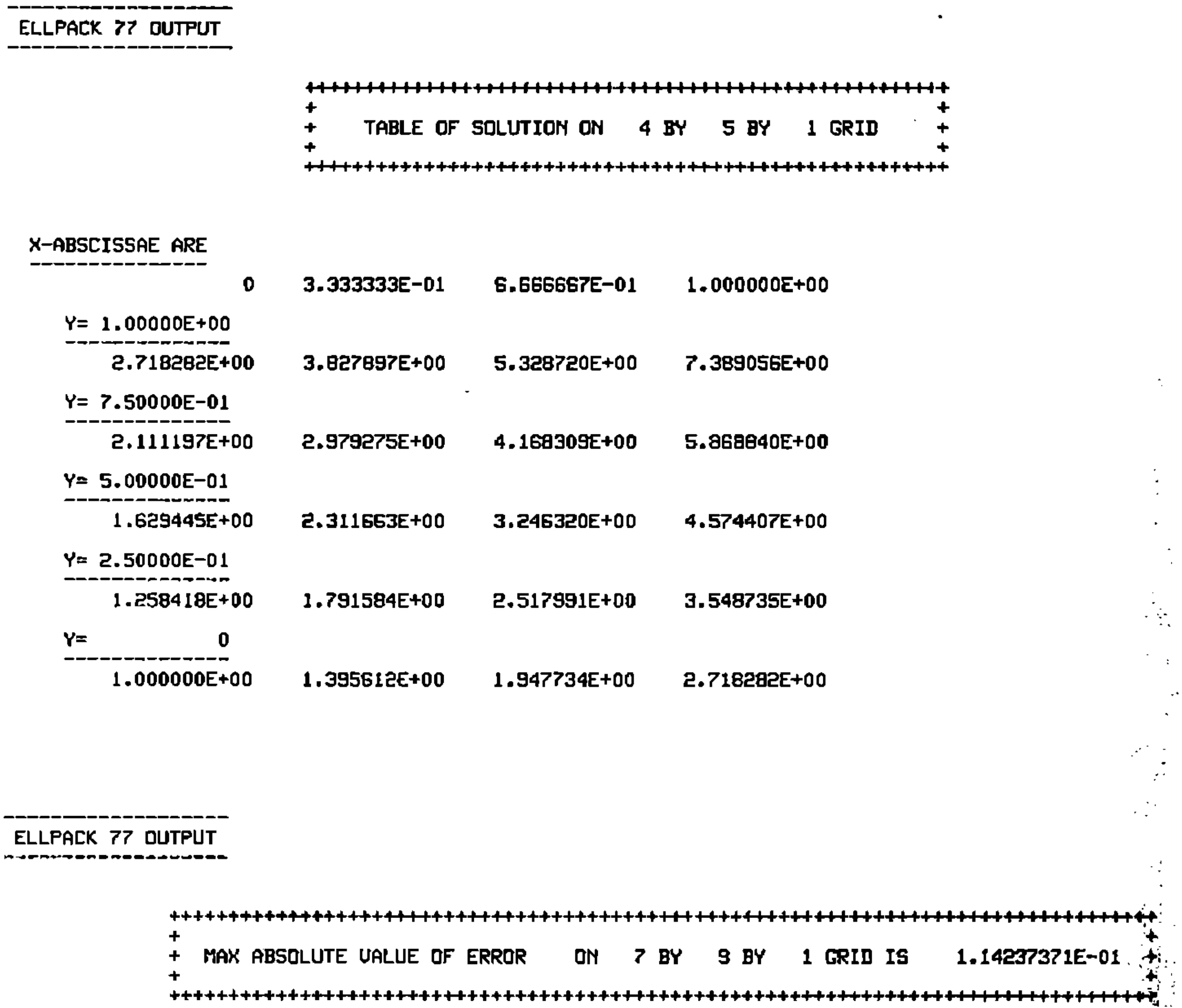
The following is the Level 1 output produced by the 5-POINT STAR, NATURAL, and BAND SOLVE modules. They correspond to the DIS, INDEX and SOL in the SEQUENCE segment of the ELLPACK input program.

```
--------------------------------------------------------------------------------
  F I N I T E   D I F F E R E N C E   M O D U L E     -     F I V E   P O I N T   S T A R
--------------------------------------------------------------------------------


      DOMAIN =  RECTANGLE    (          0, 1.0000E+00) X (          0, 1.0000E+00)

      DISCRETIZATION =  UNIFORM     4 X   5    HX= 3.3333E-01   HY= 2.5000E-01

      BOUNDARY CONDITIONS ON PIECES 1,2,3,4 ARE 3,1,3,1

      OUTPUT LEVEL = 1


      EXECUTION SUCCESSFUL
      NUMBER OF EQUATIONS =    12
      MAX NUMBER OF UNKNOWNS PER EQN = 5
```

```
--------------------------------------------------------------------------------
     I N D E X I N G   M O D U L E   -     N A T U R A L   O R D E R I N G
--------------------------------------------------------------------------------


      NUMBER OF EQUATIONS =    12
      EQUATIONS/UNKNOWNS NUMBERED IN ORDER GENERATED

      EXECUTION SUCCESFUL
```

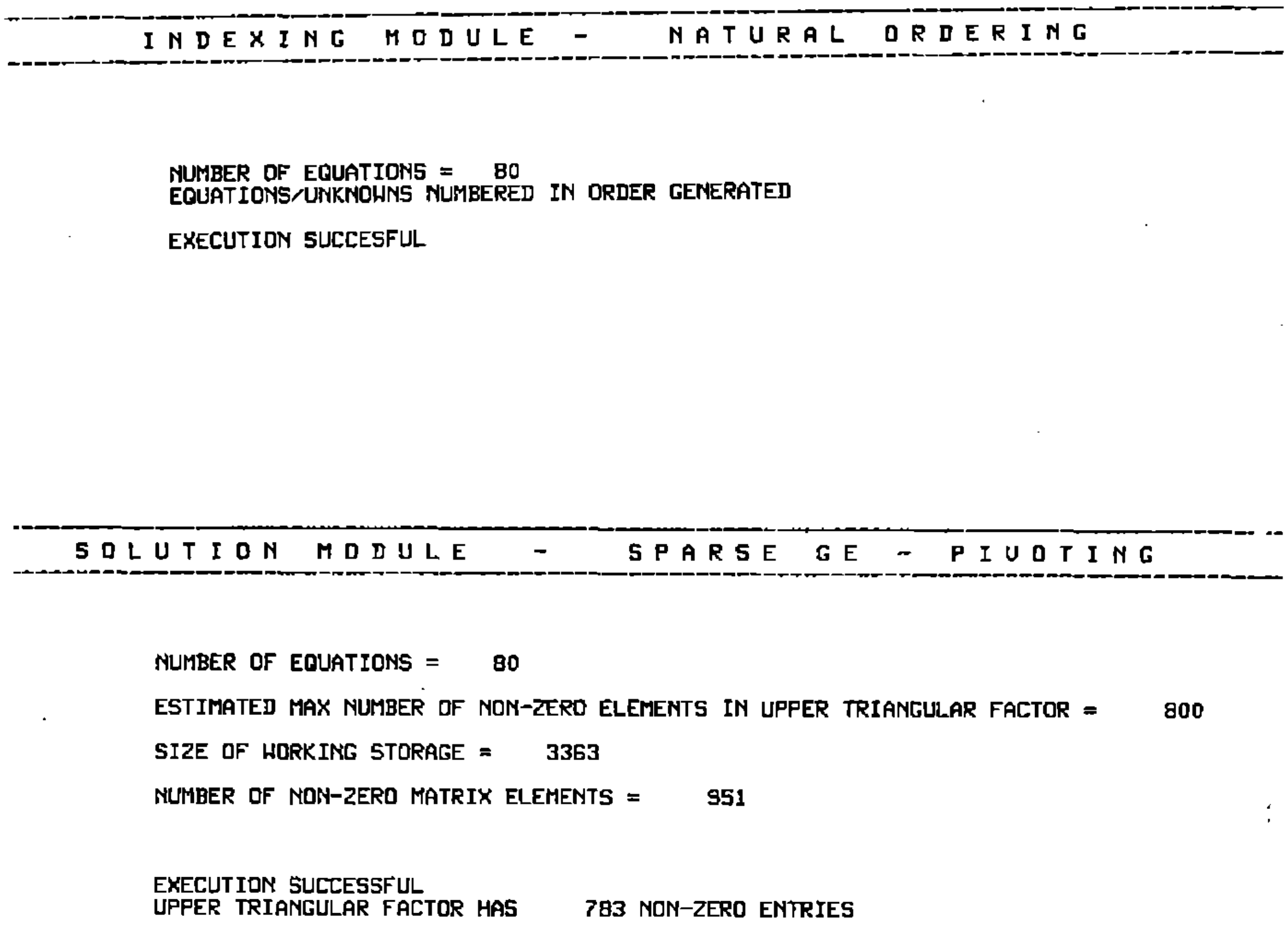
```
--------------------------------------------------------------------------------
   S O L U T I O N   M O D U L E   -   N O N S Y M M E T R I C   B A N D   S Y S T E M
--------------------------------------------------------------------------------


     NUMBER OF ROWS     =    12

     NUMBER OF COLUMNS =     9

     BANDWIDTH          =     4


     EXECUTION SUCCESSFUL
```

The following output is produced as a result of the TABLE-U and MAX(7,9)-ERROR directives in the OUTPUT(B) segment of the ELLPACK input program.



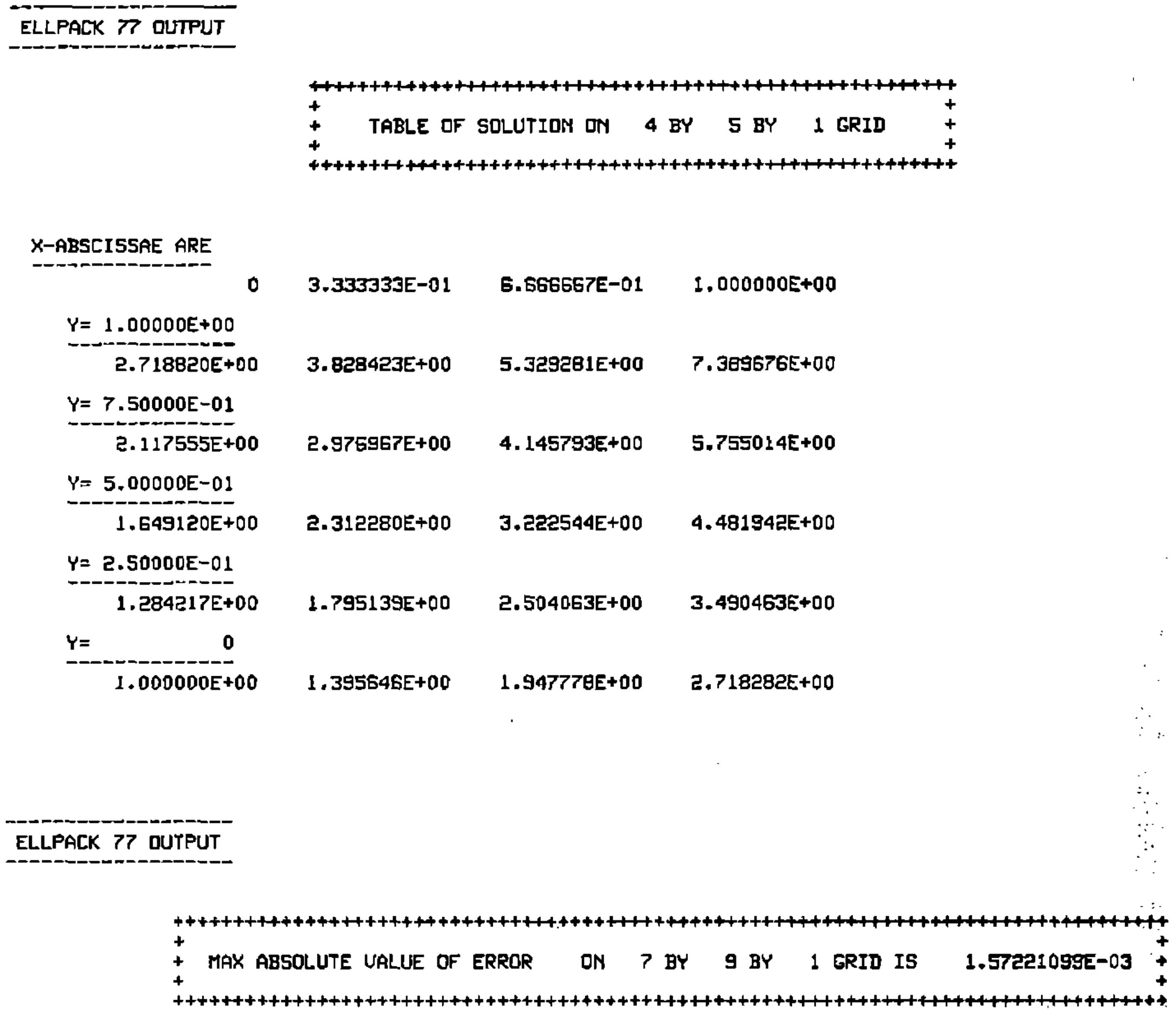
```
 ------------------
 ELLPACK 77 OUTPUT
 ------------------

                         +++++++++++++++++++++++++++++++++++++++++++++++++++++
                         +                                                   +
                         +    TABLE OF SOLUTION ON    4 BY    5 BY    1 GRID   +
                         +                                                   +
                         +++++++++++++++++++++++++++++++++++++++++++++++++++++


  X-ABSCISSAE ARE
  ---------------
                   0    3.333333E-01    6.666667E-01    1.000000E+00
     Y= 1.00000E+00
     --------------
        2.718282E+00    3.827897E+00    5.328720E+00    7.389056E+00
     Y= 7.50000E-01
     --------------
        2.111197E+00    2.979275E+00    4.168309E+00    5.868840E+00
     Y= 5.00000E-01
     --------------
        1.629445E+00    2.311663E+00    3.246320E+00    4.574407E+00
     Y= 2.50000E-01
     --------------
        1.258418E+00    1.791584E+00    2.517991E+00    3.548735E+00
     Y=            0
     --------------
        1.000000E+00    1.395612E+00    1.947734E+00    2.718282E+00



 ------------------
 ELLPACK 77 OUTPUT
 ------------------

             ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
             +                                                                            +
             +  MAX ABSOLUTE VALUE OF ERROR    ON    7 BY    9 BY    1 GRID IS    1.14237371E-01  +
             +                                                                            +
             ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

Next, the user's ELLPACK input program specifies that a second solution to the problem be attempted using the P3-C1 COLLOCATION, NATURAL and SPARSE GE-PIVOTING modules. The Level 1 output of the latter two modules (INDEX and SOL(2)) is shown below. The DIS(2) module produces no Level 1 output.



```
--------------------------------------------------------------------------------
         I N D E X I N G   M O D U L E   -     N A T U R A L   O R D E R I N G
--------------------------------------------------------------------------------


          NUMBER OF EQUATIONS =   80
          EQUATIONS/UNKNOWNS NUMBERED IN ORDER GENERATED

          EXECUTION SUCCESFUL




--------------------------------------------------------------------------------
   S O L U T I O N   M O D U L E    -    S P A R S E   G E  -  P I V O T I N G
--------------------------------------------------------------------------------


         NUMBER OF EQUATIONS =     80

         ESTIMATED MAX NUMBER OF NON-ZERO ELEMENTS IN UPPER TRIANGULAR FACTOR =      800

         SIZE OF WORKING STORAGE =     3363

         NUMBER OF NON-ZERO MATRIX ELEMENTS =       951


         EXECUTION SUCCESSFUL
         UPPER TRIANGULAR FACTOR HAS      783 NON-ZERO ENTRIES
```

Once again the OUTPUT(B) segment is invoked, this time producing the following printed output.



```
-------------------
ELLPACK 77 OUTPUT
-------------------

                    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                    +                                                          +
                    +    TABLE OF SOLUTION ON    4 BY    5 BY    1 GRID        +
                    +                                                          +
                    ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 X-ABSCISSAE ARE
 ---------------
                  0    3.333333E-01    6.666667E-01    1.000000E+00
    Y= 1.00000E+00
    --------------
        2.718820E+00   3.828423E+00    5.329281E+00    7.389676E+00
    Y= 7.50000E-01
    --------------
        2.117555E+00   2.976967E+00    4.145793E+00    5.755014E+00
    Y= 5.00000E-01
    --------------
        1.649120E+00   2.312280E+00    3.222544E+00    4.481942E+00
    Y= 2.50000E-01
    --------------
        1.284217E+00   1.795139E+00    2.504063E+00    3.490463E+00
    Y=            0
    --------------
        1.000000E+00   1.395646E+00    1.947778E+00    2.718282E+00


-------------------
ELLPACK 77 OUTPUT
-------------------

           ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
           +                                                                                    +
           +  MAX ABSOLUTE VALUE OF ERROR    ON    7 BY    9 BY    1 GRID IS    1.57221099E-03  +
           +                                                                                    +
           ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
```

The final output resulting from the execution of the ELLPACK control program is a listing of module timings. One entry appears for each segment invoked in SEQUENCE (except -- each execution of a SOLUTION module gives two timings, one for reformatting the matrix produced by the discretization and the other for the equation solution). The last time is the total execution time.

```
EXECUTION TIME FOR MODULES  (SEC.)
------------------------------

 1 TIME =    .27         2 TIME =    .08         3 TIME =    .03         4 TIME =    .03
 5 TIME =    .02         6 TIME =    .26         7 TIME =    .19         8 TIME =    .03
 9 TIME =    .09        10 TIME =    .82        11 TIME =    .39        12 TIME =   7.22
13 TIME =   9.44
```

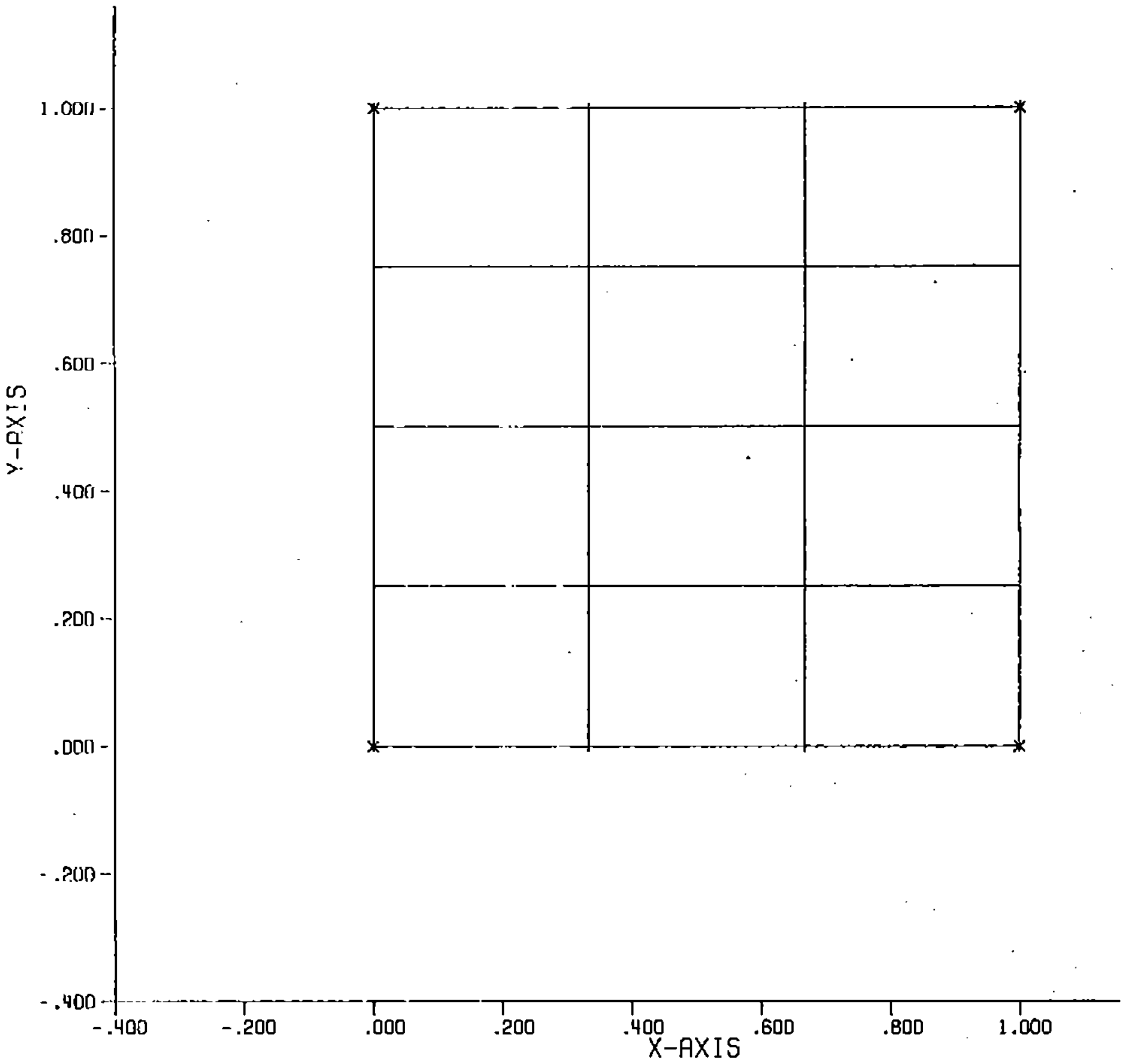
5. The SAVE File

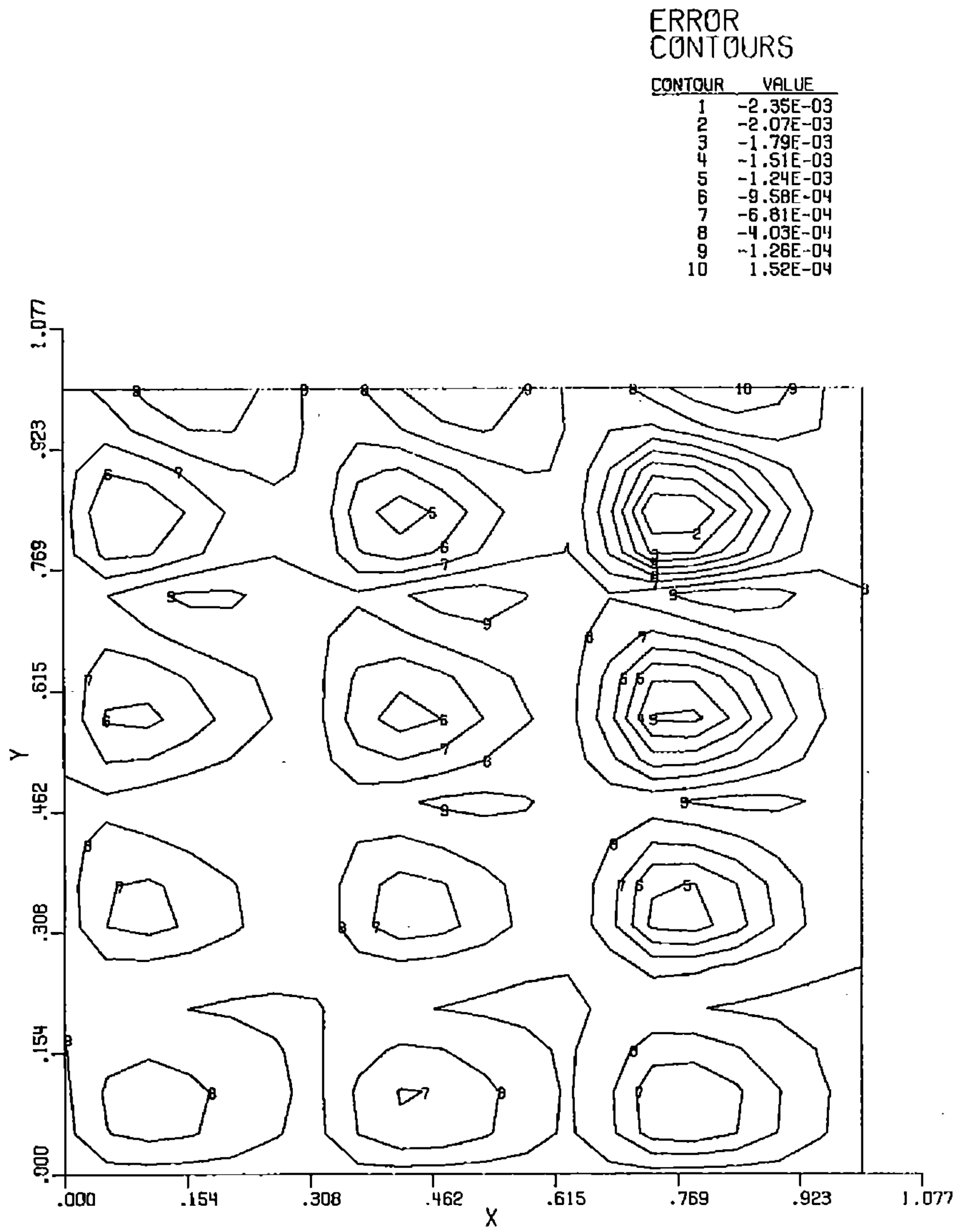
The following is the information placed on the SAVE file by the ELLPACK control program. It is produced as a result of the OUTPUT(C) segment of the ELLPACK input program.

```
------TEST-CASE-ONE-----
                         DATE = 09/14/78
   DIM2,POISON,LAPLAC,CONSTC,SELFAD,CROSST, HOMOEQ =  T  F  F  F  F  F  F
   AX,BX,AY,BY,AZ,BZ =          0 1.000000         0 1.000000        0        0
   DIRICH,NEUMAN,MIXED,HOMOBC =  T  F  T  F   BCTYPE = 3 1 3 1 X X
   UNIFRM =  T   NGRIDX,Y,Z =  4  5  1   HX,HY,HZ =  .3333333  .2500000          0
   TABLE OF SEGMENT TIMES
        1   .271  2   .082  3   .027  4   .035  5   .016  6   .264  7   .194
        8   .028  9   .093 10   .824 11   .387 12      0
    MEMORY =14234     MAX ERROR,RESIDUAL =    1.5722E-03          0
COLLOCATION-SOLUTION--------------------------
        80
 1.000000E+00 1.001023E+00 1.000492E+00 9.887249E-01 1.284217E+00 1.284882E+00
 1.283898E+00 1.282283E+00 1.649120E+00 1.649518E+00 1.648353E+00 1.647905E+00
 2.117555E+00 2.117399E+00 2.116476E+00 2.116601E+00 2.718820E+00 2.717621E+00
 2.717743E+00 2.718107E+00 1.395646E+00 1.395971E+00 1.395742E+00 1.393964E+00
 1.795139E+00 1.815774E+00 1.800977E+00 1.861500E+00 2.312280E+00 2.341061E+00
 2.334049E+00 2.419872E+00 2.976967E+00 3.002412E+00 3.020825E+00 3.097790E+00
 3.828423E+00 3.843504E+00 3.896564E+00 3.938431E+00 1.947778E+00 1.948033E+00
 1.947774E+00 1.948714E+00 2.504063E+00 2.524620E+00 2.492026E+00 2.431572E+00
 3.222544E+00 3.251347E+00 3.178390E+00 3.093429E+00 4.145793E+00 4.171397E+00
 4.057618E+00 3.982051E+00 5.329281E+00 5.344591E+00 5.192187E+00 5.143744E+00
 2.718282E+00 2.719431E+00 2.717333E+00 2.739955E+00 3.490463E+00 3.490902E+00
 3.490376E+00 3.492784E+00 4.481942E+00 4.482295E+00 4.481934E+00 4.482468E+00
 5.755014E+00 5.755340E+00 5.755014E+00 5.755356E+00 7.389676E+00 7.390126E+00
 7.389676E+00 7.390129E+00
```

6. Graphics Output

The following two plots are produced by the sample program. The first results from the PLOT-DOMIAN in OUTPUT(A) and the second from the PLOT-ERROR in OUTPUT(C). (The second plot depicts the error contours resulting from the P3-C1 COLLOCATION discretization).




ERROR
CONTOURS
CONTOUR VALUE
1 -2.35E-03
2 -2.07E-03
3 -1.79E-03
4 -1.51E-03
5 -1.24E-03
6 -9.58E-04
7 -6.81E-04
8 -4.03E-04
9 -1.26E-04
10 1.52E-04
Y
.000
.154
.308
.462
.615
.769
.923
1.077
X
.000
.154
.308
.462
.615
.769
.923
1.077

APPENDIX II

Additional Notes on Implementing ELLPACK on IBM 360/370 Systems (OS)

The following is a loose description of Figure 1 using the jargon of IBM's JCL. Many details have been omitted; our purpose is only to give a very general idea of what form the JCL must take.

As described perviously, two permanent files must be created from the ELLPACK source modules:

PREOBJ = the Preprocessor in load module form; main program called ELLPRE

ELMLIB = the ELLPACK module library with routines in object deck form

When these files have been created, the following set of control cards will implement Figure 1. We suggest that they be embodied in a catalogued procedure on your system.

Ø=oh
0=zero

```
//PREPRØS  EXEC  PGM=ELLPRE
//STEPLIB        DD DSNAME=PREØBJ,DISP=SHR
//FT01F001       DD DSN=HEADER,DISP=(NEW, PASS),UNIT=SYSDA,SPACE=(TRK,(5,5)),
//            DCB=(RECFM=FB, LRECL=80, BLKSIZE=800)
//FT08F001       DD DSN=DATA,...same as above...
//FT02F001       DD DSN=ASSIGN,...same as above...
//FT03F001       DD DSN=MØDSEQ,...same as above...
//FT04F001       DD DSN=FØRT,...same as above...
//FT06F001       DD SYSØUT=A
//FT05F001       DD *
                 ellpack input program
//ELLRØ     EXEC  FØRTGCLG
//FORT.SYSIN     DD DSN=HEADER,DISP=ØLD
//               DD DSN=DATA,DISP=ØLD
//               DD DSN=ASSIGN,DISP=ØLD
//               DD DSN=MØDSEQ,DISP=ØLD
//               DD DSN=FØRT,DISP=ØLD
//LKED.SYSLIB    DD DSN=ELMLIB,DISP=SHR
//               DD DSN=SYS1.FØRTLIB,DISP=SHR
//GØ.FT04F001    DD DUMMY*
//GØ.FT08F001    DD DSN=SCRTCH,DISP=NEW,UNIT=SYSDA,SPACE=(TRK,(5,5)),
//            DCB=(RECFM=VB,LRECL=512,BLKSIZE=517)
```

*This must be replaced if the SAVE option is implemented.