1978

ELLPACK 77 User's Guide

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

Report Number:
78-289

http://docs.lib.purdue.edu/cstech/219

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
This report describes how to use the ELLPACK 77 version of the ELLPACK system. ELLPACK 77 is a system for solving elliptic problems on rectangular domains. More information describing the objectives and structure of the ELLPACK project may be obtained from the references listed at the end. The ELLPACK 77 language is now stable and no new language features are expected to be added. Additional solution methods will be incorporated into the system.

CONTENTS

1. General Description and A Simple ELLPACK Program
2. The ELLPACK Segments
   A. General Organization of an ELLPACK Program
   B. Segments Which Define the Problem and Grid
   C. Segments Which Specify the Methods to be Used
   D. Fortran Input and Control Segments
   E. Output and Options Segments
3. Second ELLPACK Example Including Output
4. Complex ELLPACK Program for a Non-Linear PDE

REFERENCES

APPENDIX 1: Organization of the ELLPACK System
APPENDIX 2: The ELLPACK 77 Control Program for Example 3
APPENDIX 3: The ELLPACK 77 Modules (as of 6/15/80)
   A. Summary List
   B. Module Descriptions
1. GENERAL DESCRIPTION.

The ELLPACK language is a problem statement language for elliptic partial differential equations (PDEs) in two or three independent variables. The ELLPACK system implements this language using a library of Fortran subprograms. ELLPACK 77 is the version which is restricted to PDEs on rectangular domains. The subprograms are grouped together to form modules for specific tasks in the solution process. For example, a DISCRETIZATION module forms a system of linear equations which approximates the PDE problem and a SOLUTION module solves the system. Several choices are available for each type of module and one can be selected in the ELLPACK language by simply giving its name.

ELLPACK was originally developed as a research and educational tool to evaluate mathematical software for solving elliptic problems. Contributions to the ELLPACK system have been made by researchers at several institutions.*

A problem is specified by an ELLPACK program written in the simple user oriented ELLPACK language. The ELLPACK system processes this program by first converting it to a Fortran source program which includes, for example, specific array declarations. Then the system compiles this source program and links it to precompiled ELLPACK subprograms and finally executes it.

Figure 1 shows a very simple problem and an ELLPACK program which generates an approximate solution. A table of this solution is printed and a contour plot is produced; the actual output is shown in Figure 2.

*These include: Harvard University, G. Birkhoff; Purdue University, R. Boisvert, R. E. Lynch, T. Papatheodorou, J. R. Rice (coordinator); University of Texas at Austin, R. Banks, R. Grimes, D. Kincalid, A. Sherman, D. M. Young; University of South Carolina, E. Houstis; University of New Mexico, C. Moler.
Problem:

Estimate the solution of

\[ u_{xx} + u_{yy} + 3u_x - 4u = \exp(x+y)\sin(x), \ 0 < x < 1, -1 < y < 2, \]

subject to the Dirichlet boundary conditions

\[ u = 0, \quad x = 0, \ -1 \leq y \leq 2, \]
\[ u = x, \quad 0 \leq x \leq 1, \quad y = 2, \]
\[ u = y/2, \quad x = 1, \ -1 \leq y \leq 2, \]
\[ u = \sin(\pi x) - x/2, \quad 0 \leq x \leq 1, \quad y = -1. \]

Use the divided central difference approximation (5-POINT STAR) to approximate the PDE at mesh points of a square mesh with spacing 1/5.

**ELLPACK program:**

```plaintext
EEEQTOOON Dimensions $ CONSTANT COEFFICIENTS
UXX$ + UYY$ + 3.0UX$ - 4.0U$ = EXP(X+Y)*SIN(X)

BNDARY. $ X = 0.0, \ U = 0.0
Y = -1.0, \ U = SIN(3.1415926536*K) - K/2.0
X = 1.0, \ U = Y/2.0
Y = 2.0, \ U = X

GRID. $ UNIFORM X = 8 $ UNIFORM Y = 16

DJS. 5-POINT STAR
INDEXING. NATURAL
SOLUTION. LINPACK BAND
OUTPUT. TABLE-SOLUTION $ PLOT-SOLUTION
OPTIONS. TIME $ MEMORY

END.
```

Figure 1. Sample problem and ELLPACK program.
DISCRETIZATION MODULE

5-POINT STAR

DOMAIN
X INTERVAL 0, 1.000E+00
Y INTERVAL -1.000E+00, 2.000E+00
DISCRETIZATION GRID
HY 2.000E-01
HY 2.000E-01
B.C.S ON PIECES 1,2,3,4 1,1,1,1
OUTPUT LEVEL 1
NUMBER OF EQUATIONS 56
MAX NO. OF unknowns per eq. 5
EXECUTION SUCCESSFUL

INDEXING MODULE

NATURAL

NUMBER OF EQUATIONS 56
EQUATIONS/unknowns numbered in order generated
EXECUTION SUCCESSFUL

SOLUTION MODULE

LINPACK BAND

NUMBER OF ROWS 13
NUMBER OF COLUMNS 56
NUMBER OF LOWER CO-DIAGONALS 4
NUMBER OF UPPER CO-DIAGONALS 4
LINPACK BAND GIVES 2 TIMINGS
SETUP TIME AND SOLUTION TIME
EXECUTION SUCCESSFUL

ELLPACK 77 OUTPUT

+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
+ TABLE OF SOLUTION ON 5 X 16 GRID +
+ + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + + +
X-ABSCISSAE ARE

<table>
<thead>
<tr>
<th>0</th>
<th>2.000000E-01</th>
<th>4.000000E-01</th>
<th>6.000000E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.000000E-01</td>
<td>1.000000E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Y = 2.000000E+00

<table>
<thead>
<tr>
<th>4.400852E-16</th>
<th>2.000000E-01</th>
<th>4.000000E-01</th>
<th>6.000000E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.000000E-01</td>
<td>1.000000E+00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Y = 1.800000E+00

<table>
<thead>
<tr>
<th>1.110223E-15</th>
<th>1.201611E-01</th>
<th>2.231957E-01</th>
<th>3.482890E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.431435E-01</td>
<td>9.000000E-01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 2. The ELLPACK output for the sample problem. Each module prints a simple summary message, then the requested table occurs followed by the timings as requested in the OPTIONS. The final time is the total, the others are from the individual modules (the LINPACK BAND module has two timings). The plotted output is also given.
The program consists of several segments whose names (EQUATION, BOUNDARY, and so on) begin in column 1 of a card, the rest is written in free format excluding column 1. Each segment includes a selection of key-words (2 DIMENSIONS, CONSTANT COEFFICIENTS, and so on) which supply the system with information about the problem. The dollar sign is a separator. Parts of the program include Fortran expressions (+3., EXP(X+Y)*SIN(X), etc.) which must follow the rules of Fortran.

This example is the simplest case of an ELLPACK program; one defines the PDE problem in the EQUATION and BOUNDARY segments; the rectangular grid is defined in the GRID segment, the method of solution is chosen in the DISCRETIZATION, INDEXING and SOLUTION segments, the desired output is specified in the OUTPUT segment and OPTIONS are chosen. Every ELLPACK program ends with END.

The ELLPACK language is specified in the next section, then two more complicated examples are presented (one solves the same problem with two different choices of methods, the other solves a non-linear PDE by iteration on approximate linear PDEs). The appendices contain a one page description of each module available, an overview of the ELLPACK system and an example of an ELLPACK Control Program. The latter two items are of interest to people who wish to try more complicated things such as the non-linear PDE in Example 3.

2. **THE ELLPACK SEGMENTS**

2.A. **GENERAL ORGANIZATION OF AN ELLPACK PROGRAM.** The basic blocks of statements in an ELLPACK program are segments. Some segments must appear exactly once and others may appear more than once in which case each one must have a unique identifier in parentheses as in DISCRETIZATION(1) and
DISCRETIZATION (A2). A brief summary of the segments is given below:

**Group 1** segments must appear exactly once and before any from Group 2:

- **EQUATION.** Specifies the PDE equation.
- **BOUNDARY.** Specifies the rectangular domain and the boundary conditions.
- **GRID.** Specifies the vertical and horizontal mesh lines.

**Group 2** segments may appear more than once as desired:

- **DISCRETIZATION.** Specifies a method used to approximate the PDE problem; this generates a system of linear algebraic equations.
- **INDEXING.** Specifies an ordering of the algebraic equations and the unknowns.
- **SOLUTION.** Specifies a method for solving the algebraic system.
- **OUTPUT.** Specifies a desired ELLPACK generated output (printed and graphical).

**Group 3** segments may appear anywhere in the program and as many times as desired:

- ***** Specifies a comment. Cannot be inside other segments.
- **OPTIONS.** Specifies which of various options are desired.
- **FORTRAN.** Specifies that the statements which follow make up user supplied Fortran subprograms.

**Group 4** segments can appear at most once after all Group 2 segments and the OPTIONS segments.

- **SEQUENCE.** Specifies the sequence of execution of the Group 2 segments and the number of times they are executed.

A simple looping facility is provided and a call on a user supplied subroutine TEST may be made to
perform postprocessing, updating, etc. If this optional segment is omitted, then the modules are executed in the order they are given in the ELLPACK program.

END. The required last statement which specifies the end of the ELLPACK program and directs the ELLPACK system to execute it.

All segment names and their abbreviations must end with a period; if a Group 2 segment is repeated, it must have an identifier in parentheses as indicated above. Each DISCRETIZATION., INDEXING., SOLUTION., and OPTIONS. segment must be on a single line, that is, the key-words associated with these segment names are put on the same line as the names. At most 999 non-blank characters can make up a single segment.

In an ELLPACK program, blanks are ignored, but blank lines are not allowed. Dollar signs are used to separate statements and keywords. The end of a line (card) is also a separator and if a statement requires more than a single line, then it is continued to subsequent lines by placing a period in column 1.

The independent variables are denoted by X and Y for two-dimensional problems and by X, Y, and Z for three-dimensional ones. The dependent variable is denoted by U, its first derivatives $\frac{\partial U}{\partial X}$, $\frac{\partial U}{\partial Y}$, and $\frac{\partial U}{\partial Z}$ by UX, UY, and UZ, and its second derivatives UXX, UYY, UXY, and so on.

We now give the key-words, other information and examples for each of the segment types.

2.3 SEGMENTS WHICH DEFINE THE PROGRAM AND GRID (GROUP 1). We describe these segments by first listing their key-words and their meanings. These key-words are to be used by the modules or the ELLPACK system; modules are not obligated to act on the information supplied by key-words. For example,
a DISCRETIZATION module might not take advantage of the fact that the user has specified that a simple Laplace's equation is to be discretized. We then describe the rules (syntax) for writing the segments and give a number of examples.

**EQUATION.** segment (abbreviation EQ.).

The EQUATION. segment specifies the differential equation to be solved. The segment name is followed by key-words separated by $; as many as appropriate should be used to give the system as much information about the equation as possible; this might be used to simplify the solution process.

The key-words are:

- **POISSON**: The PDE is the Poisson equation.
- **LAPLACE**: The PDE is the Laplace equation.
- **CONSTANT COEFFICIENTS**: The differential operator has constant coefficients.
- **HOMOGENEOUS**: The right side of the equation is zero.
- **SELF-ADJOINT**: The PDE is in self-adjoint form. In the ELLPACK language, the equation
  \[(p(x,y)u_x)_x + (q(x,y)u_y)_y + r(x,y)u = s(x,y)\]
  is written as
  \[P(X,Y) UXX$ + Q(X,Y) UYY$ + R(X,Y) U = S(X,Y)\]

- **TWO DIMENSIONS**: There are two independent variables, X and Y.
- **THREE DIMENSIONS**: There are three independent variables, X, Y and Z.

In the definition of the equation, the dependent variable and its derivatives are denoted by

- \(U\), \(UX\), \(UY\), \(UXX\), \(UYY\), \(UZZ\), \(UXY\), and \(UXZ\). If the coefficient of one is zero, then it does not appear. The dollar signs merely make it easy for the system to locate the terms in the PDE. They are required except just in front of the equal sign. The equation itself is specified in the
'operator' = 'right side'

where 'operator' is a list of terms of the form

'coefficient' 'derivative' or 'coefficient' * 'derivative'

The 'coefficient' and 'right side' denote any valid Fortran real arithmetic expressions as well as + and -, and 'derivative' denotes one of U$, UX$, and so on.

Some examples of the EQUATION. segment are given below.

EQUATION. LAPLACE $ 2 DIMENSIONS

UX$ + UYY$ = 0

EQUATION. CONSTANT COEFFICIENTS

TWO DIMENSIONS

-4. UX$ + .377 UXY$ - 10.2 UYY$ + 3.E+4*UX = SIN(X+COS(X*Y))

EQUATION. 2 DIMENSIONS $ HOMOGENEOUS

(X**2 + Y**2 + 16.)*UX$ + VALUYY(X,Y)*UY$ + 2.234E-3*ATAN2(Y,X) UX$ + 1.~*UY$ - VALU(X,Y)*U$ = 0.

* IN THE PRECEDING EQUATION, THE COEFFICIENTS OF UYY AND U ARE
* GIVEN AS FUNCTIONS, THESE MUST BE SUPPLIED BY THE USER IN A FORTRAN.
* SEGMENT

EQUATION. 2 DIMENSIONS

COFX(X,Y)*UX$ + COFY(X,Y)*UY$ + COFX(X,Y) UX$ + COFY(X,Y)*UY$ = COFRS(X,Y)

* IN THIS EQUATION, ALL OF THE COEFFICIENTS ARE USER SUPPLIED FORTRAN
* FUNCTIONS

EQUATION. 2 DIMENSIONS $ SELF-ADJOINT

* EXAMPLE OF POLAR COORDINATES, X FOR & AND Y FOR THETA

X UX$ + 1./X UTY$ - X*(4. + COS(Y)) U = X**3*SIN(2.*Y)
EQUATION.  2 DIMENSIONS $ POLSSON

UXX $ + UYY $ = EXP ( ESTU(X,Y) )

* USED TO ESTIMATE SOLUTION OF UXX $ + UYY $ = EXP( U ) BY PICARD
* ITERATION. AN INITIAL ESTIMATE OF U IS CONSTRUCTED. VALUES OF
* THE PREVIOUS ITERATE ARE RETURNED BY THE USER SUPPLIED FUNCTION ESTU.
* THE USER SUPPLIED SUBROUTINE TEST TRANSMITS THE PREVIOUS ITERATE
* JUST COMPUTED TO ESTU.

Remarks: Except for SELF-ADJOINT, the key-words are optional. Some of the modules take advantage of special features of a problem to reduce execution time and they can do this only if the key-words are used. Some modules check keywords to validate their applicability. Thus, a method that requires constant coefficients might not work on UXX$ + UYY=0 unless the keyword CONSTANT COEFFICIENTS is given.

BOUNDARY. segment (abbreviation BOUND.).

The BOUNDARY. segment specifies the sides of the rectangular boundary and the boundary conditions on them. The key-words are:

DIRICHLET   The value of U is given on all four sides.
NEUMANN     The problem has a non-unique solution.
MIXED       Mixed conditions are given; this is also used
             in the boundary conditions specification.
HOMOGENEOUS The right sides of all the boundary conditions
             are zero.

A side and the condition is specified by

'variable' = 'constant' , 'condition'

where 'variable' is one of X, Y, or Z, 'constant' is a Fortran expression which evaluates to a constant, and 'condition' is one of the following:

U = 'expression'
UX or UY or UZ = 'expression'
MIXED = ('exp') U ('exp') UX ('exp') UY ('exp') UZ = 'expression'
where 'expression' and 'exp' denote Fortran real arithmetic expressions and if one of 'exp' is zero, then it, the enclosing parentheses, and the derivative which follows can be left off. The order of the terms in the MIXED 'condition' does not matter.

If NEUMANN is specified, then the user must provide in a Fortran segment the function TRUE(X,Y) or TRUE(X,Y,Z) (even if the exact solution is not known), the ELLPACK system calls this function to obtain a boundary value for \( U \) in order to make the solution unique. The point used for this is \((AX,AY)\) or \((AX,AY,AZ)\) where \( Av \) is the smaller of the two values which give the locations of the sides of the region in the v-direction. If the PDE has Neumann boundary conditions but the solution is nevertheless unique (e.g. for \( u_{xx} + u_{yy} + u = 1 \)) then one of the boundary conditions must be written in the MIXED format.

Some examples of BOUNDARY. segments follow.

BOUNDARY. DIRICHLET $ HOMOGENEOUS $ \( X = 1., U = 0. \) $ X = 10., U = 0. $ X = -1., U = 0. $ Y = -10., U = 0.

BOUNDARY.

\[
X = 10., \ UX = 0. \ \text{\$ Y = 4.\*ATAN(1.), U = 0. \$ X = -1., U = 0. \ \text{\$ Y = 2.\*EXP(1.), U = 0.}}
\]

\[
Y = 0., \ UX = 0. \ \text{\$ Y = -2*X + 3.12}}
\]

* 4.\*ATAN(1.) EVALUATES TO \( \pi \) IF YOU FORGET THE DIGITS OF \( \pi \), EXP(1.) = e.

BOUNDARY. \( X = 0., U = 0. \ $ Y = 0., \ \text{\UY = -2*X + 3.12}}

\[
X = 1., \ \text{MIXED = (1.) U (-3.) UX = EXP(Y)}
\]

\[
Y = 1., \ \text{MIXED = ( VALU(X) ) U ( VALUY(X) ) UY = VALUS(X)}}
\]

* THE FORTRAN FUNCTIONS VALU, VALUY, VALUS must appear in a Fortran segment.

BOUNDARY. \( X = -1., \ \text{MIXED = (B(X,Y,1,1))U (B(X,Y,1,2))UX = B(X,Y,1,4) \)

\[
Y = -1., \ \text{MIXED = (B(X,Y,2,1))U (B(X,Y,2,2))UX (B(X,Y,2,3))UY = B(X,Y,2,4)}}
\]

\[
X = 2., \ \text{MIXED = (B(X,Y,3,1))U (B(X,Y,3,2))UX (B(X,Y,3,3))UY = B(X,Y,3,4)}}
\]

\[
Y = 3., \ \text{MIXED = (B(X,Y,4,1))U (B(X,Y,4,2))UX (B(X,Y,4,3))UY = B(X,Y,4,4)}}
\]
FORTRAN.

FUNCTION B(X,Y,IEQN,ITEM)

B = 0.

GO TO ( 1000, 2000, 3000, 4000 ), IEQN

1000 CONTINUE

C BOUNDARY CONDITION ALONG X = -1.
IF( ITEM .EQ. 1 ) B = Y**2 + COS ( EXP( -2.*Y ) )
IF( ITEM .EQ. 2 .AND. Y .GT. 0. ) B = Y**4
IF( ITEM .EQ. 4 ) B = EXP( SIN(Y) )

GO TO 5000

2000 CONTINUE

C OUTWARD DIRECTED NORMAL EQUAL TO +1/4. ALONG Y = -1.
IF( ITEM .EQ. 3 ) B = 1.
IF( ITEM .EQ. 4 ) B = -1/4.

GO TO 5000

3000 ...

5000 CONTINUE
RETURN
END

GRID. segment (no abbreviation)

The GRID. segment defines the rectangular grid placed in the domain. There are no independent key-words, the specification of a uniform grid is made by

UNIFORM variable = number

where variable is X, Y or Z and number is the number of uniformly spaced grid lines in this variable including the domain edges.

For a non-uniform grid we have

NGRID variable = number, list
where variable is X, Y or Z, number is the number of grid lines and list contains the values (separated by commas) of the variable at all the grid lines. The values in the list must be constants.

Some examples of GRID. segments follow.

GRID. UNIFORM X = 5 $ UNIFORM Y = 10
GRID. UNIFORM X = 5 $ NGRID Y = 12, 0.0, 0.123, 0.24689, 0.479532, 0.6978401, 0.80823, 1.0, 1.35794, 1.864329, 2.09094, 2.46785, 3.0
* THE PERIOD IN COLUMN 1 OF THE SECOND LINE CONTINUES THE LIST OF Y GRID LINES.
GRID. NGRID X = 5, 0.0, 0.2, 0.5, 0.8, 1.0 $ NGRID Y = 4, 0.0, 0.3, 0.7, 1.0
UNIFORM Z = 4

2.C. SEGMENTS WHICH SPECIFY THE METHODS TO BE USED. We next describe the three segments which specific methods (that is, particular library modules) to be used in the solution process; these are DISCRETIZATION, INDEXING and SOLUTION.

Some modules accept parameters in parenthesis after the name such as in FFT 9-POINT(IORDER=4). A more detailed description of all of these modules is given in Appendix 3. Most EILPACK programs have all three of these segments present corresponding to the three steps in approximately solving the PDE. However, some modules incorporate all three of these steps; they are listed in DISCRETIZATION but they would not be followed by an INDEXING or SOLUTION module. When multiple methods are used, the sequencing of the modules is specified by the SEQUENCE. segment described in Section 2.D.

DISCRETIZATION. segment (abbreviation DIS.)

This segment names a module to be used to form a linear system of equations. The content of this segment is a single module name and the
list of available modules is expandable. The first six modules listed below actually carry out all the steps in solving the PDE. One page descriptions are given in Appendix 3 for each module.

Modules that Discretize and Solve

DYAKANOV-CG Conjugate gradient iteration for second order approximation to self-adjoint PDE.

DYAKANOV-CG4 Fourth order version of preceding module.

FFT 9-POINT High order fast fourier transform for 2D, constant coefficients.

HODIE 27-POINT 3D High order $O(h^6)$ finite difference for 3D Poisson problem.

MARCHING ALGORITHM Fast algorithm for separable, self-adjoint problems.

2DEP Galerkin with quadratic, triangular elements.

Modules that Only Discretize

5-POINT STAR Ordinary finite differences in 2 dimensions.

7-POINT STAR Ordinary finite differences in 3 dimensions.

HODIE-ACF High order finite differences for $Au_{xx} + Cu_{yy} + Pu$

HODIE-HELMHOLTZ High order finite differences for the Helmholtz problem

P3-Cl COLLOCATION Collocation on rectangular mesh with cubic-Hermite elements

P3-Cl GALERKIN Galerkin on rectangular mesh with cubic-Hermite elements

INDEXING. segment (abbreviation INDEX.)

The modules specified in this segment take the linear system produced by the DISCRETIZATION. and reorganize it by renumbering the equations and/or unknowns. For example, one may wish to have the nested dissection ordering of the equations before using Gauss-elimination to solve them.

NESTED DISSECTION Computes the nested dissection ordering of the equations.

YALE RCM The Reverse Cuthill-McKee algorithm is applied to minimize the band width of the linear system.
NATURAL. The natural ordering is that of the generation of the equations and unknowns by the discretization modules. This module does practically nothing.

RED-BLACK The variables and unknowns are numbered as on a checker board, all "red" points before the "black" points.

YALE MIN DEG Matrix rows and columns are reordered to give minimal degree.

SOLUTION. segments (abbreviation SOL.)

The modules specified by this segment actually solve the linear systems of equations. This step may also involve reformatting the equations. For example, BAND SOLVE requires the equations to be in a certain band matrix format before the Gauss elimination is done. This module does the reformatting as well as the solution of the equations.

BAND SOLVE Gauss elimination for a band matrix

JACOBI CG Jacobi iteration with conjugate gradient acceleration

JACOBI SI Jacobi iteration with semi-iterative acceleration

LINPACK BAND Gauss elimination for a band matrix

LINPACK SPD BAND Gauss elimination for a symmetric positive definite band matrix

REDUCED SYSTEM CG Reduced system iteration with conjugate gradient acceleration

REDUCED SYSTEM SI Reduced system iteration with semi-iterative acceleration

SOR SOR iteration

SPARSE GE-PIVOTING Sparse Gauss elimination with pivoting

SSOR CG SSOR iteration with conjugate gradient acceleration

SSOR SI SSOR iteration with semi-iterative acceleration

SYMMETRIC BAND Gauss elimination for a symmetric band matrix

YALE SPARSE Four variations of sparse matrix Gauss elimination
Examples of method specification segments follow.

* ORDINARY FINITE DIFFERENCES AND GAUSS ELIMINATION

DISCRETIZATION. 5-POINT STAR
INDEXING. NATURAL
SOLUTION. BAND SOLVE

* A FINITE ELEMENT METHOD WITH ITERATION

DISCRETIZATION. P3G1-GALERKIN
INDEXING. NATURAL
SOLUTION. SOR

* A SINGLE MODULE FOR THE 3 SEGMENTS

DIS. FFT 9-POINT(ORDER=4)

* MODULES TO SOLVE THE SAME PROBLEM BY GAUSS ELIMINATION:

DIS. 5-POINT STAR
INDEX(1). NATURAL
INDEX(2). RED-BLACK
SOLUTION(A). YALE SPARSE(DRV=S, NSF=18000)
SOL(B). JACOBI SI(TMAX=200, ZETA=1.E-4)
SOL(C1). SSOR SI(TMAX=200, ZETA=1.E-4, CMR=.82)
SOL(C2). REDUCED SYSTEM SI(TMAX=200, ZETA=1.E-4, IADAPT=1)

It is important to note that not all combinations of modules are legal. Some illegal combinations are fairly obvious such as we cannot apply SYMMETRIC BAND SOLVE to a general PDE which will not have a symmetric linear system approximation. Similarly, an INDEXING module to minimize matrix band width is not likely to work with an iterative module for solving the equations. The module descriptions in Appendix 3 indicate which combinations are legal, however, there are so many combinations...
possible that one should be cautious when first using a particular one. The table at the beginning of Appendix 3 gives some guidance as to which combinations are legal.

2.D. FORTRAN INPUT AND CONTROL SEGMENTS. The basic use of the FORTRAN segment is to define various functions used in the problem. These may be coefficients of the PDE or boundary conditions or perhaps the true solution if known. The ELLPACK system generates a Fortran Control Program from the ELLPACK language input and the Fortran in the FORTRAN segment is just passed along to be compiled with the Control Program.

A more complicated use of the Fortran segment is to interact with the Control Program in solving the problem. All of the important variables of the Control Program are in labeled COMMON blocks and thus they are available to a user's Fortran program. This requires considerable familiarity with the Control Program and the operation of the ELLPACK system. An example Control Program is given in Appendix 2 and the ELLPACK system is described in Appendix 1. More detailed information on this aspect can be obtained from the references at the end of this report.

The basic use of the SEQUENCE segment is to specify the order in which the modules are to be executed. This segment may also interact with the Fortran programs provided in order to attempt more difficult problems. The non-linear PDE of Section 4 is such an example.

FORTRAN segment (abbreviation F.).

The ELLPACK system generates Fortran programs based on the user's ELLPACK program. The statements are written onto a file. The statements in the FORTRAN segments are copied onto this file which is then compiled, linked to the precompiled subprograms in the ELLPACK library, and then the
resulting object code is executed.

The FORTRAN segments contain user supplied Fortran subprograms with user made up names. These may contain input or output statements (the ELLPACK system uses standard READ and WRITE units which at most installations are 5 and 6, e.g. READ(5,label) ... and WRITE(6,label) ...).

Warning: The user's Fortran subprogram names must not be the same as any ELLPACK Fortran names. Violating this can cause mysterious behavior.

In these subprograms, column 1 must either be blank or contain a C or $. A statement label 10000 or greater will cause an error. Some Fortran systems require certain control cards for each Fortran program. These may be inserted with a $ in column 1; the $ is stripped off and the card contents moved one column to the left.

When one is testing the accuracy of a discretization method, one often solves a problem with known solution. With ELLPACK, one can then compute the error, tabulate it, plot it and so on. The function subprogram TRUE is reserved for evaluation of the exact solution of the problem and it must be supplied by the user if the error is to be determined:

FUNCTION TRUE(X,Y) or FUNCTION TRUE(X,Y,Z).

At installations which use a double precision version of ELLPACK, one uses DOUBLE PRECISION FUNCTION TRUE(X,Y) and similarly for other user supplied functions.

If the key-word TEST appears in the SEQUENCE segment, then the user must supply a subroutine with this name and a single integer argument in one of the FORTRAN segments:

SUBROUTINE TEST(LOOP).

The value of LOOP is reduced by one after each execution of TEST and the run is terminated when LOOP is equal to zero.
One can initialize on the first call of a routine as well as use TEST to solve a set of problems as indicated below.

FORTRAN.

FUNCTION RIGHTS(X,Y)

C EVALUATES THE RIGHT SIDE OF THE DIFFERENTIAL EQUATION

COMMON / COMRGH / AMPLID, PERIOD

CALL INITIL(2)

RIGHTS = AMPLID*SIN((X+Y)/PERIOD)

RETURN

END

SUBROUTINE INITIL(ISET)

C READ THEAMPLITUDE AND PERIOD IF IFIRST IS 1

COMMON / COMRGH / AMP, PER

DATA IFIRST / 1 /

IF(ISET.EQ.1) IFIRST = 1

GO TO (100, 200), IFIRST

100 CONTINUE

READ(5,110) AMP, PER

110 FORMAT(2E20.12)

WRITE(6,120) AMP, PER

120 FORMAT(// 6X, 50(1H/) / 6X, 21HAMPLITUDE AND PERIOD=, 2E20.12/ 6X, 50(1H/) //)

IFIRST = 2

200 CONTINUE

RETURN

END

SUBROUTINE TEST(LOOP)

C RESET IFIRST IN INITIL

CALL INITIL(1)

RETURN

END
The Fortran programs of the user can access the computed solution $U(X,Y)$ or $U(X,Y,Z)$ of the PDE with the function subprogram given below. In particular, the program TEST discussed in the SEQUENCE segment (see page 22) can use $U$ to carry out some kind of "past processing" of the numerical solution obtained. This function $U$ is automatically available as part of the ELLPACK system.

```
FUNCTION U(X, Y, Z)
*** THIS FUNCTION SUPPLIES THE PDE SOLUTION FROM
*** ELLPACK. NONSTANDARD ARRAY DIMENSIONS ARE USED;
*** ACTUAL DIMENSIONS MAY BE OBTAINED FROM THE ELLPACK
*** CONTROL PROGRAM IF NEEDED.
DIMENSION IROT(6)
COMMON / INTEGS / NUMBD, NUMCOE, NROW, NCOL, NBAND,
A IROT, LEVEL, INITL, INDIS, INSOL,
B MINPUT, MOUTPT, MEMORY, INING, NGRDXZ,
C NGRIDZ, NGRDZ, MAXEQL, MAXCOZ, NROWDZ,
D NGRIDZ
COMMON / COEF2Z / COEF(I, 1)
COMMON / IDC02Z / IDCOEF(I, 1)
COMMON / MDX2Z / MDX(I, 1)
COMMON / MDXU2Z / MDX(I, 1)
COMMON / INUXZ / INUX(I, 1)
COMMON / UNKN2Z / UNKN(I)
COMMON / GRIDZ2 / GRIDX(I, 1)
COMMON / GRIDY2 / GRIDY(I)
COMMON / GRIDZ2 / GRIDZ(I)
COMMON / TABL2Z / TABLEM(I, 1, 1)
COMMON / IGRIZZ / IGRID(I)
COMMON / BCTYZZ / BCTYPE(I)
U = SOLUX(X, Y, Z, NGRDXZ, NGRDZ,
A NGRIDZ, GRIDX, GRIDY, GRIDZ, UNKN,
B MAXEQL, MAXUNK, TABLEM, COEF, IDCOEF,
C MAXCOZ, BCTYPE, INUX(I, MDX(I, IGRID))
RETURN
END
```
SEQUENCE. segment (abbreviation SEQ.).

This segment allows the user to specify the order for executing DISCRETIZATION, INDEXING, SOLUTION and OUTPUT segments in an ELLPACK run. Either the full or abbreviated names of the segments may be used. Modules of the same type are distinguished by adding tags to the ends of the segment names; the tags are one or two characters in parentheses at the end of the segment names as in DIS(1), DIS(X), OUTPUT(A4), and so on; a period must follow the right parenthesis when the segment is defined, the period is optional in the SEQUENCE segment.

In addition, a simple loop can be created, beginning with LOOP = 'constant,' where 'constant' is a Fortran integer constant, and ending with the key-word 'TEST,' which is the last item in the SEQUENCE list. If 'TEST' is not used, it may be omitted. The action of the loop is as indicated in the following:

```
LOOP = 'constant'
100 CONTINUE
execute the ELLPACK segments listed
CALL TEST(LOOP)
LOOP = LOOP - 1
IF( LOOP .GT. 0 ) GO TO 100
```

The key-words in the SEQUENCE. segment are:

- LOOP: LOOP = 'constant' creates a simple loop. Only one loop may appear and it goes to the end of the SEQUENCE list.
The optional last statement of the SEQUENCE list specifies a call to the user supplied subroutine TEST(LOOP).

The execution of the SEQUENCE and the ELLPACK run are terminated as by a normal Fortran STOP statement; control is passed to the next control card. This is useful for complex applications where one uses output from one ELLPACK run as input for another ELLPACK run or for additional processing, saving, and so on. It is also useful for making partial runs while debugging complex or time consuming applications.

Some examples of the SEQUENCE segment follow.

* The USUAL SEQUENCE

SEQUENCE. DISCRETIZATION $ INDEXING $ SOLUTION $ OUTPUT

* SEQUENCING THE MODULES OF THE LAST EXAMPLE OF SECTION 2.C

SEQUENCE. DIS $ INDEX(1) $ SOLUTION (A) $ OUTPUT

$ SOLUTION (B) $ OUTPUT

$ SOL (C1) $ OUTPUT

$ INDEX(2) $ SOL (C2) $ OUTPUT

* A SIMPLE LOOP

SEQUENCE. DIS $ LOOP=4 $ INDEXING $ SOLUTION $ OUTPUT

* USE OF TEST AS A POST-PROCESSOR OF THE ELLPACK RUN

SEQ. DIS $ INDEX $ SOL $ OUT $ TEST

* USE OF TEST TO TERMINATE A LOOP

SEQ. DIS $ INDEX(1) $ SOL(1) $ OUTPUT

LOOP=100 $ INDEX(2) $ SOL(2) $ OUTPUT $ TEST
2.E. OUTPUT AND OPTIONS SEGMENTS.

OPTIONS segment (abbreviation OPT.)

This segment may be used to select various features of the ELLPACK system. OPTIONS segments may be at most one line long, although they may be repeated as many times as desired. In complicated applications it may be necessary to place some options at the start of the ELLPACK program, they must always precede the SEQUENCE segment.

NO EXECUTION Do not run ELLPACK program
STOP Terminate preprocessor (useful only for debugging it)
LEVEL Set output levels (0-5) in ELLPACK run
LEVEL=0 Requests no output from modules except fatal error messages
LEVEL=1 Requests minimal output
LEVEL=2 Requests reasonable summary of what happened
LEVEL=3,4,5 More and more intermediate output, primarily useful for debugging
DEBUG Set output levels (0-5) in the ELLPACK language preprocessor, when DEBUG is set to 2 or more the preprocessor will process multiple ELLPACK programs without putting any of them into execution.
TIME Give the execution times of each module. Some modules may produce more than one timing—see the detailed descriptions in Appendix 3.
MEMORY Give estimates of the memory used in the ELLPACK run with some breakdown.
INITIAL

Provide an initial value for the UNKNOWN array (the actual unknowns of the linear system generated by the discretization module). Useful for iterative methods and nonlinear or time dependent applications. The syntax is INITIAL = 'name' where user provides

SUBROUTINE 'name' (U,NU)

REAL U(NU)

to initialize the UNKNOWN array. INITIAL may be used for any kind of initializing computation the programmer finds useful.

OLDU

Provide previously computed UNKNOWN arrays. Useful for iterative methods and nonlinear or time dependent applications. The syntax is OLDU = (Integer) where the value of the (Integer) is the number of previous U's to provide.

The values of the current and previous unknowns may be accessed in a user's Fortran by including the following statements

COMMON / INTEqs / NUMBEQ,NDNUM(16)
COMMON / UNKNZ / UNKNOWN(1)
COMMON / OLDU1 / UNKNOWN(1)
COMMON / OLDU2 / UNKNOWN(1)

where NUMBEQ is the actual size of the arrays and more OLDU's may have been specified.

Some OPTIONS. example segments follow.

* BASIC STATISTICS ON PERFORMANCE

OPTIONS. TIME $ MEMORY

* INITIALIZE UNKNOWNS AND DEFINE 2 PREVIOUS ONES

OPT. TIME $ MEMORY $ LEVEL=1

OPT. INITIAL = GUESS1 $ OLDU=2
OUTPUT segment (abbreviation OUT.)

This segment specifies various kinds of output from the computation.

The key-words are compound of the form

verb - noun

and the list of verbs and nouns is:

Verbs

MAX Print maximum value on the grid
MAX(NX,NY,NZ) Print maximum value on the equispaced grid defined by NX, NY and NZ
PLOT Contour plot of a function of two variables or plot of the domain with grid lines
TABLE Tabulate at grid points
TABLE(NX,NY,NZ) Tabulate on the equispaced grid defined by NX, NY and NZ
SAVE(Message) Save on disk or other user storage (with Message heading) the PARAMETERS (of problem and timing) or U (=SOLUTION) in the FORMAT (ILO/(6E13.5)) where the integer is the length of U.

Nouns

TRUE True solution given by user
SOLUTION or U Solution just computed
ERROR TRUE - U
RESIDUAL Computed residual of equation. Self-adjoint case requires functions CDXU(X,Y) and CDYU(X,Y) giving coefficients of UX, UY in non-self-adjoint form of equation.
DOMAIN The problem domain and grid lines
PARAMETERS The key-word attributes of the segments EQ., BOUND., GRID, plus other problem defining quantities and the list of execution times. All are labeled as in the example later.
INDEXES
The three indexing arrays NDXEQ, NDXUNK, INVNIX of the Control Program (only valid with TABLE).

UNKNOWN
The UNKNOWN array (only valid with TABLE) of the linear equation problem generated by DISCRETIZATION.

Some OUTPUT example segments follow:

* TEST HOW WELL A METHOD DOES
OUTPUT.
MAX-ERROR $ MAX-RESIDUAL $ TABLE-U $ PLOT-ERROR

* SAVE THE SOLUTIONS FOR SOME LATER USE
* THE SAVED INFORMATION IS AS SHOWN IN FIGURE 4-B
OUTPUT(S).
SAVE (EXAMPLE--2-- PARAMETERS) - PARAMETERS
SAVE-U

3. SECOND ELLPACK EXAMPLE INCLUDING OUTPUT.

This example is more complicated than the first and it shows how one can compare two distinct methods for solving the same problem. In this PDE the ordinary 5-POINT STAR finite differences achieves an error of .115 in .16 seconds. The finite element method P3-CI COLLOCATION achieves an accuracy of .0016 in 1.13 seconds. A much finer grid would be required for 5-POINT STAR to achieve an accuracy of .0016 and the time required would be larger than 1.16 second.

The timings can be identified as follows:

1 - OUT(A)  2 - DIS  3 - INDEX  4 & 5 - SOL  6 - OUTPUT(B)
7 - DIS(2)  8 - INDEX  9 & 10 - SOL(2) 11 - OUTPUT(B) 12 - OUT(C)
13 - total time
Figure 3. The second ELLPACK example program. This is one of the standard test problems for the ELLPACK system.
**DISCRETIZATION MODULE**

### 5-Point Star

<table>
<thead>
<tr>
<th>Domain</th>
<th>Rectangle</th>
</tr>
</thead>
<tbody>
<tr>
<td>X Interval</td>
<td>0, 1.000E+00</td>
</tr>
<tr>
<td>Y Interval</td>
<td>0, 1.000E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discretization</th>
<th>Uniform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid</td>
<td>4 x 5</td>
</tr>
<tr>
<td>Hx</td>
<td>3.333E-01</td>
</tr>
<tr>
<td>Hy</td>
<td>3.333E-01</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B.C.s on Pieces</th>
<th>1, 2, 3, 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Level</td>
<td>1</td>
</tr>
</tbody>
</table>

| Number of Equations | 12         |
| Max No. of Unknowns Per Eq. | 5          |

**EXECUTION SUCCESSFUL**

---

**INDEXING MODULE**

### Natural

<table>
<thead>
<tr>
<th>Number of Equations</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equations/Unknowns Numbered in Order Generated</td>
<td>12</td>
</tr>
</tbody>
</table>

**EXECUTION SUCCESSFUL**

---

**SOLUTION MODULE**

### LINPACK Band

<table>
<thead>
<tr>
<th>Number of Rows</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Columns</td>
<td>12</td>
</tr>
<tr>
<td>Number of Lower Co-Diagonals</td>
<td>4</td>
</tr>
<tr>
<td>Number of Upper Co-Diagonals</td>
<td>4</td>
</tr>
</tbody>
</table>

**LINPACK Band Gives 2 Timings**

**Setup Time and Solution Time**

**EXECUTION SUCCESSFUL**

---

**ELLPACK 77 OUTPUT**

++++++

<table>
<thead>
<tr>
<th>X-ABSCISSAE ARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 3.333333E-01</td>
</tr>
<tr>
<td>Y = 1.000000E+00</td>
</tr>
<tr>
<td>2.718282E+00</td>
</tr>
<tr>
<td>Y = 7.389056E+00</td>
</tr>
<tr>
<td>1.629445E+00</td>
</tr>
<tr>
<td>Y = 2.500000E-01</td>
</tr>
<tr>
<td>1.258418E+00</td>
</tr>
<tr>
<td>Y = 0</td>
</tr>
<tr>
<td>1.000000E+00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y = 1.355612E+00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y = 1.547734E+00</td>
</tr>
<tr>
<td>Y = 2.718282E+00</td>
</tr>
</tbody>
</table>

**EXECUTION SUCCESSFUL**

---

**ELLPACK 77 OUTPUT**

++++++

| MAX(ABSERROR ) ON 7 X 9 GRID = 1.1423737E-01 |

**EXECUTION SUCCESSFUL**

---

**Figure 4-A. Continued.**
DISCRETIZATION MODULE

P 3 - C 1 COLLOCATION

CASE NONHOMGENEOUS
DOMAIN RECTANGLE
X INTERVAL 0, 1.000E+00
Y INTERVAL 0, 1.000E+00
GRID 4 X 5
HX 3.333E-01
HY 2.500E-01
OUTPUT LEVEL 1
NUMBER OF EQUATIONS 80
MAX NO. OF UNKNOWNS PER EQ. 16
EXECUTION SUCCESSFUL

INDEXING MODULE

NATURAL

NUMBER OF EQUATIONS 80
EQUATIONS=UNKNOWNs NUMBERED IN ORDER GENERATED
EXECUTION SUCCESSFUL

SOLUTION MODULE

SPARSE GE-PIVOTING

NUMBER OF EQUATIONS 80
ESTIMATED MAX. NUMBER OF NON-ZERO ELEMENTS IN UPPER TRI. FACTOR 900
SIZE OF WORKING STORAGE 3363
SP. GE-PIV. GIVES 2 TIMINGS
SETUP TIME AND SOLUTION TIME.
NUMBER OF NON-ZERO MATRIX ELEMENTS 951
NUMBER OF NON-ZERO ENTRIES IN UPPER TRIANGULAR FACTOR 763
EXECUTION SUCCESSFUL

ELLPACK 77 OUTPUT

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ TABLE OF SOLUTION ON 4 X 5 GRID +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

X-ABSCISSAE ARE
---------------- 0 3.333333E-01 6.666667E-01 1.000000E+00
Y = 1.000000E+00
---------------- 2.718281E+00 3.828427E+00 5.000000E+00
Y = 7.500000E-01
---------------- 2.117555E+00 2.978697E+00 4.147933E+00 5.180563E+00
Y = 2.500000E-01
---------------- 1.595627E+00 2.255277E+00 3.135833E+00 4.115982E+00
Y = 5.000000E-01
---------------- 1.209036E+00 1.788109E+00 2.361439E+00 3.075603E+00
Y = 1.000000E+00
-------------- 1.000000E+00 1.395646E+00 1.795139E+00 2.604063E+00
EXECUTION SUCCESSFUL

-----------------------------------
+ MAX(ABS(ERROR)) ON 7 X 9 GRID = 1.5722110E-03 +
+-----------------------------------

Figure 4-A. Continued.
**ELLPACK 77 OUTPUT**

+++++++ EXECUTION TIMES +++++++

<table>
<thead>
<tr>
<th>MODULE NAME</th>
<th>SECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTPUT</td>
<td>.266</td>
</tr>
<tr>
<td>5-POINT STAR</td>
<td>.066</td>
</tr>
<tr>
<td>NATURAL</td>
<td>.008</td>
</tr>
<tr>
<td>INTERFACE SETUP</td>
<td>.018</td>
</tr>
<tr>
<td>LINPACK BAND</td>
<td>.029</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>.247</td>
</tr>
<tr>
<td>P2-C1 COLLOCATION</td>
<td>.205</td>
</tr>
<tr>
<td>NATURAL</td>
<td>.009</td>
</tr>
<tr>
<td>INTERFACE SETUP</td>
<td>.073</td>
</tr>
<tr>
<td>SPARSE GE-PIUING</td>
<td>.006</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>.357</td>
</tr>
<tr>
<td>OUTPUT</td>
<td>7.137</td>
</tr>
<tr>
<td>TOTAL EXECUTION</td>
<td>9.225</td>
</tr>
</tbody>
</table>

**Figure 4-A.** The printed output from the program of Figure 3.

---TEST-CASE-ONE-----

<table>
<thead>
<tr>
<th>DATE: 09/19/78</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIM2, POISON, LAPLAC, CONST, SELFAD, CROSST, HOMEOQ = T F F F F T F</td>
</tr>
<tr>
<td>AX, BX, HY, BY, AZ, BZ = 0 1.000000 0 1.000000 0 0</td>
</tr>
<tr>
<td>DIRICH, NEUMANN, MIXED, HOMOBC = T F F F MCTYPE = 3 1 3 1 X X</td>
</tr>
<tr>
<td>UNIFORM = T FINDIM, V, Z = 4 5 1 HX, HY, HZ = .333333 .250000 0</td>
</tr>
<tr>
<td>TABLE OF SEGMENT TIMES</td>
</tr>
<tr>
<td>1 .270 2 .083 3 .028 4 .035 5 .015 6 .266 7 .193</td>
</tr>
<tr>
<td>8 .028 9 .087 10 .013 11 .374 12 0</td>
</tr>
<tr>
<td>MEMORY =14234 MAX ERROR, RESIDUAL = 1.5722E-03 0</td>
</tr>
</tbody>
</table>

**Figure 4-B.** The information saved on an external file by the SAVE statements.
Figure 4-C. The graphical output from PLOT-DOMAIN. This facility is useful for non-rectangular domains in ELLPACK 78.
Figure 4-D. The graphical output from OUT(G) which includes PLOT-ERROR. The contours shown are typical of the errors for collocation solutions.
TRUE = EXP(X*Y)
RETURN
END
SUBROUTINE TESTC!

***** MONITOR CHANGE IN SOLUTION AT END OF EACH ITERATION

REAL DIFF(50)
COMMON / UNKNWZ / UNKNW(I)
COMMON / INTEGS / NUMEQ, NNUMB(I5)
COMMON / OLDUI / UNKNW(I)

ITER = 5 - 1
WRITE(G,1010) ITER
DIFMAX = 0.0
DO 10 J=1,NUMEQ
  DIFF(J) = UNKNW(J) - UNKNW(J)
  DIFMAX = AMAX(DIFMAX,ABS(DIFF(J)))
10 CONTINUE
WRITE(G,1020) DIFMAX, (J, DIFF(J), J=1,NUMEQ)
WRITE(G,1030)
RETURN
1010 FORMAT(/ 1HO, 5X, 20(I4*), 2X, 9HITERATION, I4, 2X, 10(2H**))
1020 FORMAT(1HO, 5X, 25HMAX CHANGE IN SOLUTION = , E12.5, 3X, A 15H DIFFERENCES ARE // 4(1H , 4(I5, E11.2)) )
1030 FORMAT(1HL)
END
SUBROUTINE SETUP(U,N)

***** INITIALIZE SOLUTION AND INDEXING VECTORS

COMMON / NDXUZ2 / NDXUNK(I)
REAL U(N)

DO 10 1=1,N
  FI = I*0.001
  U(I) = 1.0 + FI
  NDXUNK(I) = I
10 CONTINUE
RETURN
END

* END.

------------------------
PREPROCESSOR OUTPUT
------------------------

APPROXIMATE MEMORY REQUIREMENTS

| WORKSPACE  | 4004 | GRID LINES | 13 |
| LIN EQ COEFS | 160  | LIN EQ ID-S | 160 |
| INDEXES OF VARS | 48   | AMATRIX, BUECTR | 112 |
| PROGRAM + MISC | 2716 | TOTAL MEMORY | 7213 |

ELLPACK PREPROCESSOR TIME .88 SECONDS

Figure 5 - Continued
The arrays NDUMB(7), RDUMB(6) and NDUMB(15) are dummies to make the lengths of the COMMON blocks right. Clearly, writing an ELLPACK program like this requires considerable understanding of how the ELLPACK system and Control Program work.

A complicated problem involving modifying the domain is described in reference [2] below.

REFERENCES


2. Lynch, R. E., P. Gherson and P. S. Lykoudis, The use of ELLPACK 77 for solving the Laplace equation on a region with interior slits, application to a problem in magnetohydrodynamics, CSD-TR 275, Computer Science Department, Purdue University, July 1978.


4. __________, ELLPACK 77 Contributor's Guide, CSD-TR 267, Computer Science Department, Purdue University, June 10, 1978.


6. __________, ELLPACK 78 User's Guide, CSD-TR 306, Computer Science Department, Purdue University, Revised July 1, 1980.


APPENDIX ONE - ORGANIZATION OF THE ELLPACK SYSTEM

The ELLPACK system processes a PDE problem in the following steps:

1. Read and interpret the ELLPACK program. This is done by a preprocessor which is a Fortran program. That is, the ELLPACK program is data read into this preprocessor. The output from the preprocessor is a Fortran program—the ELLPACK Control Program. Appendix two contains an example of the output.

2. The ELLPACK Control Program is compiled.

3. The compiled ELLPACK Control Program is loaded along all required modules. The modules are in a library and already compiled.

4. Execution of the ELLPACK run. The ELLPACK Control Program plus the modules apply the numerical methods to the PDE problem and produce the requested output.

The schematic chart of the processing of an ELLPACK run is given in Figure A1.
Figure A1. Schematic diagram of the processing of an ELLPACK run. This report describes the user's input language for ELLPACK 77.
An example of the output of the ELLPACK preprocessor is given in Appendix 2.

Another view of the structure of an ELLPACK 77 computation is illustrated in Figure A2.

![Diagram of ELLPACK 77 computation]

**Figure A2.** Basic organization of an ELLPACK 77 computation. The user specifies the modules to be used and more than one combination may be used on a single ELLPACK run.

The user defines the PDE problem to be solved and the rectangular grid to be used. He then specifies the combination of modules to be used in the solution. Note that some modules actually span more than one pair of interfaces. For example, FFT9 starts at Interface 1 and stops at Interface 4.
APPENDIX TWO - SAMPLE OUTPUT FROM PREPROCESSOR

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
FILE HEADER FOR ELLPACK 77
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

1. 000000B PROGRAM ELL77 (INPUT, OUTPUT, PLOT, SAVE, SCRATCH, TAPE4=SAVE,
   A TAPES=INPUT, TAPE6=OUTPUT, TAPE8=SCRATCH)

2. 005145B DIMENSION IROTS(6)

3. 005145B REAL
   AX, BY, AZ,
   BZ, MX, HY, MZ,
   CUXY, CUY, CUM, CUZ,
   CUKZ, CUYZ, CUZZ, CUEZ, EPSGRD
   GRIDX, GRIDY, GRIDZ, AMATRX, BUECTR
   UNKNWN, COEF, TABLEM, BPARAN, XBOUND

4. 005145B INTEGER
   IROT, MORZ, VERT, BOTH, INTER,
   PERIOD, CORNER, BLANK, JUMP

5. 005145B LOGICAL
   DIM2, DIM3, POISON, LAPLAC, CONST,
   SELFD, CROSS, DIRICH, NEUMAN, MIXED,
   UNIFRM, DEBUG, TIMER, HOMEOG, HOMOBC
   SYMNET, FATAL, CLOCKW, RECTAN

6. 005145B COMMON WORKS

7. 005145B COMMON / PROBL / DIM2, DIM3, POISON, LAPLAC, CONST,
   SELFD, HOMEOG, CROSS, HOMOBC, DIRICH,
   NEUMAN, MIXED, UNIFRM

8. 005145B COMMON / PROBR / AX, BX, BY, AZ,
   BZ, MX, HY, MZ,
   CUXY, CUY, CUM, CUZ,
   CUKZ, CUYZ, CUZZ, CUEZ

9. 005145B COMMON / PROB1 / GRIDX, GRIDY, GRIDZ

10. 005145B COMMON / CPDE / CUXX, CUXY, CUY, CUM, CUZ

11. 005145B COMMON / CONTROL / DEBUG, TIMER, SYMNET, FATAL, RECTAN

12. 005145B COMMON / INTEGS / NUMBEQ, NUMCAE, NROW, NCOL, NBAND

13. 005145B COMMON / REALS / ERRMAX, SOLMAX, TRUMAX, RESMAX, TIMES

14. 005145B COMMON / BNDRY / NPIECE, MBOUND, NBOUND, CLOCKW

15. 005145B COMMON / SYMCON / MORZ, VERT, BOTH, INTER, STAR,

16. 005145B COMMON / NUMCON / EPSGRD

17. 005145B COMMON / NUMCON1 / NBOUND, NBOUND, ARC

18. 005145B COMMON / SYM2 / JUMP

19. 005145B COMMON / COEF(22) / COEF

20. 005145B COMMON / COEF(22) / INCOEF

21. 005145B COMMON / MXE22 / MIXED

22. 005145B COMMON / MXK22 / MIXED

23. 005145B COMMON / NUMAX / NUMAX

24. 005145B COMMON / AMAT22 / AMATRX

25. 005145B COMMON / BUECTR22 / BUECTR

26. 005145B COMMON / UNKN22 / UNKNWN

27. 005145B COMMON / GRID22 / GRIDX

28. 005145B COMMON / GRID22 / GRIDY

29. 005145B COMMON / GRID22 / GRIDZ

30. 005145B COMMON / TABL22 / TABLEM

31. 005145B COMMON / IGRID22 / IGRID

32. 005145B COMMON / BANGE22 / BRANGE

33. 005145B COMMON / BTYPE22 / BCTYPE

34. 005145B COMMON / GTYPE22 / GTYPE

35. 005145B COMMON / XBOU22 / XBOUND

36. 005145B COMMON / YBOU22 / YBOUND

37. 005145B COMMON / PIEC22 / PIECE

38. 005145B COMMON / BTYPE22 / BTYPE

39. 005145B COMMON / BNE122 / BNEIGH

40. 005145B COMMON / BPARAN1 / BPARAN

41. 005145B COMMON / G22 / GRID

42. 005145B DIMENSION GRIDX(6)

43. 005145B DIMENSION GRIDY(6)

44. 005145B DIMENSION GRIDZ(1)

45. 005145B DIMENSION MOBSQ1(12)

46. 005145B DIMENSION WORKSP( 4004)

47. 005145B DIMENSION COEF( 16,10)

48. 005145B DIMENSION IDCOEF( 16,10)
A2-2

49. 0051458  \text{DIMENSION} NXEQ(16), MUNK(16), MUNK(16), INUNK(16), AGRID(16)

50. 0051458  \text{DIMENSION} NMATR(6, 16), MUNK(16), MUNK(16), BUCCTR(16)

51. 0051458  \text{INTEGER} LTYPE(6), IGRID(16)

52. 0051458  \text{EXTERNAL} SOLUT, ERROR, RESIN.

53. 0051458  \text{INTEGER} GTYPE(1, 1)

54. 0051458  \text{REAL} BRANGE(2, 1)

55. 0051458  \text{INTEGER} PIECE(1), BPTYPE(1), BNEIGH(1)

56. 0051458  \text{DIMENSION} XEOUNO(1), YBOUND(1)

57. 0051458  \text{INTEGER} PIECE(1), BPTYPE(1), BNEIGH(1)

58. 0051458  \text{DIMENSION} TABLEMC(6, 1)

59. 0051458  \text{REAL} UNKW(1)

60. 0051458  \text{COMMON} / OLDU/ UNKNW(1)

61. 0051458  \text{INTEGER} NAMES(4, 2)

62. 0051458  \text{REAL} TIMES(37), NAMES(37)

63. 0051458  \text{DATA} IGRID: IQURET, IQUETH, IQINTEGR, IQUPERI, IQUGAM, IQPLAN

64. 0051458  \text{DATA} LGRID: IQURET, IQUETH, IQINTEGR, IQUPERI, IQUGAM, IQPLAN

65. 0051458  \text{DATA} LGRID: IQURET, IQUETH, IQINTEGR, IQUPERI, IQUGAM, IQPLAN

66. 0051458  DIM2 = .TRUE.

67. 0051458  DIM3 = .FALSE.

68. 0051458  POISON = .FALSE.

69. 0051458  LAPLAC = .FALSE.

70. 0051458  CONSTC = .FALSE.

71. 0051458  SELFAD = .FALSE.

72. 0051458  HOMEQ = .FALSE.

73. 0051458  HOMDEC = .FALSE.

74. 0051458  CROSTK = .FALSE.

75. 0051458  RECTAN = .FALSE.

76. 0051458  UNIFRM = .FALSE.

77. 0051458  DIRICH = .TRUE.

78. 0051458  NEUMAN = .FALSE.

79. 0051458  MIXED = .FALSE.

80. 0051458  DEBUG = .FALSE.

81. 0051458  TIMER = .FALSE.

82. 0051458  MTRUPT = 5

83. 0051458  MDTRUPT = 6

84. 0051458  DO 5 I=1, 6

85. 0051458  5 IROT(I) = 0

86. 0051458  0053058  IROT(1) = 2

87. 0051458  0053058  IROT(3) = -2

88. 0051458  0053058  LEVEI = 1

89. 0051458  0053058  XNUX = 1

90. 0051458  0053058  CLOUW = .TRUE.

91. 0051458  0053058  SYMMET = .FALSE.

92. 0051458  0053058  AER = 0.0

93. 0051458  0053058  B2 = 0.0

94. 0051458  0053058  ERRAX = 0.0

95. 0051458  0053058  RESAX = 0.0

96. 0051458  0053058  N2 = 0.0

97. 0051458  0053058  MUXRY = 0

98. 0051458  0053058  TFIRST = 0.0

99. 0051458  0053058  TFLST = 0.0

100. 0053208  0053208  KTIMES = 0

101. 0053208  0053208  FATAL = .FALSE.
Another view of the structure of an ELLPACK 77 computation is illustrated in Figure A2.

The user defines the PDE problem to be solved and the rectangular grid to be used. He then specifies the combination of modules to be used in the solution. Note that some modules actually span more than one pair of interfaces. For example, FPT9 starts at Interface 1 and stops at Interface 4.
102. 005322B  TIME = 0.0
103. 005322B  ARC = .FALSE.
104. 005322B  NIBND = 1
105. 005325B  NIBMT = 1
106. 005325B  HORIZ = IQHORZ
107. 005326B  VERT = IQVERT
108. 005330B  BOTH = IQBOTH
109. 005331B  INTER = IQINTER
110. 005333B  STAR = IQSTAR
111. 005334B  ONE = IQONE
112. 005336B  BLANK = IQBLANK
113. 005337B  MINUS = IQMINUS
114. 005341B  XXX = IQXXX
115. 005342B  EXTER = IQEXTER
116. 005344B  PERIOD = IQPERIOD
117. 005345B  CORNER = IQCORNER
118. 005347B  JUMP = IQJUMP
119. 005350B  EPSGRD = 0.00000001

****END OF DEFAULT INITIALIZATION****

C + END OF DEFAULT INITIALIZATIONS +
C ++++++++++++++++++++++++++++++++++++++++

120. 005352B  TIMER = .TRUE.
121. 005353B  LEVEL = 0
122. 005353B  CONSTD = .TRUE.
123. 005354B  CUXX = 1.0
124. 005355B  CUXY = 0.0
125. 005356B  CUYX = 1.0
126. 005357B  CUY = 0.0
127. 005358B  CX = 0.0
128. 005359B  RECTAN = .TRUE.
129. 005360B  AX = 0.0
130. 005361B  BX = 1.0
131. 005362B  IF ( AX .LT. BX ) GO TO 210
132. 005364B  ASAUE = AX
133. 005366B  ASAVE = AX
134. 005367B  AX = BX
135. 005368B  BX = ASAVE
136. 005371B  IROT(1) = 0
137. 005372B  IROT(3) = 0
138. 005372B  210 CONTINUE
139. 005373B  AY = 0.0
140. 005373B  BY = 1.0
141. 005374B  IF ( AY .LT. BY ) GO TO 220
142. 005375B  ASAUE = AY
143. 005377B  AY = BY
144. 005378B  BY = ASAUE
145. 005401B  IROT(2) = +2
146. 005402B  IROT(4) = -2
147. 005403B  220 CONTINUE
148. 005404B  IR = 1 + IROT(1)
149. 005405B  BCTYPE(IR) = 1
150. 005406B  IR = 2 + IROT(2)
151. 005410B  BCTYPE(IR) = 1
152. 005411B  IR = 3 + IROT(3)
153. 005413B  BCTYPE(IR) = 1
154. 005414B  IR = 4 + IROT(4)
155. 005416B  BCTYPE(IR) = 1
156. 005417B  NGRIDX = 6
157. 005418B  NGRIDY = 6
158. 005421B  HX = (BX-AX)/ 5.
159. 005424B  DO 11 J = 1, 5
160. 005426B  11 GRIDX(J) = AX + FLOAT(J-1)*HX
161. 005432B  GRIDX( 6) = BX
162. 005433B  NGRIDX = 6
163. 005434B  NGYRD = 6
164. 005435B  HY = (BY-AY)/ 5.
165. 005437B  DO 12 J = 1, 5
166. 005441B  12 GRIDY(J) = AY + FLOAT(J-1)*HY
167. 005447B  GRIDY( 6) = BY
168. 005448B  UNIFRM = .TRUE.
A2-5

169. 005450B  NGRDZD = 1
170. 005451B  NGRIDZ = 1
171. 005451B  MODSQ(1) = 1
172. 005452B  MODSQ(2) = 1
173. 005452B  MODSQ(3) = 3
174. 005453B  MODSQ(4) = 4
175. 005455B  MODSQ(5) = 5
176. 005456B  MODSQ(6) = 7
177. 005457B  MODSQ(7) = 1
178. 005460B  MODSQ(8) = 2
179. 005461B  MODSQ(9) = 3
180. 005462B  MODSQ(10) = 5
181. 005463B  MODSQ(11) = 6
182. 005464B  MODSQ(12) = 6
183. 005464B  NPDIM = 4
184. 005465B  NBDIM = 1
185. 005466B  MXNCOE = 10
186. 005467B  MXNEQ = 16
187. 005470B  NGRID = 6
188. 005471B  NROWZ = 6
189. 005472B  NCOLD = 16
190. 005473B  NCOLDZ = 16
191. 005473B  NBRAND = 5
192. 005474B  MEMORY = 7213
193. 005475B  DO 10 J=1, MXNEQ
194. 005476B    NGRID(J) = -10001
195. 005476B    NDXUNK(J) = -10001
196. 005476B    INUNDX(J) = -10001
197. 005476B    IGRID(J) = -10001
198. 005478B  10 CONTINUE
199. 005479B    DO IS J=1, 37
200. 005479B    TIMES(J) = 0.0
201. 005479B  15 CONTINUE
202. 005510B  CALL SECOND(TSTART)
203. 005512B  CALL PLOTS
204. 005513B  CALL SETUP (UNKNOWN, MXNEQ)
205. 005515B  KSEG = 0
206. 005515B  GO TO 600

C
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++1
C ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++1
C ++++++++++++++++++++++++++++ FILE SEQUENCE FOR ELLPACK ?? ++++++++++++++++++
C ++++++++++++++++++++++++++++ FILE SEQUENCE FOR ELLPACK ?? ++++++++++++++++++
C ++++++++++++++++++++++++++++ FILE SEQUENCE FOR ELLPACK ?? ++++++++++++++++++

207. 005515B  500 CONTINUE
208. 005517B  IF (TIMER ) CALL SECOND(TFIRST)
209. 005520B  INIL = 1
210. 005522B  INDIS = 13
211. 005523B  METHOD = 41
212. 005524B  CALL HOMELM(NGRIDX, NGRIDY, GRIDY, NGRIDY, COEF,
213. 005524B       AX, MXNCOE, MXNEQ, IDCOEF, IGRID, BCTYPE, METHOD, ETA,
214. 005524B       B, NSET, S1, S2, S3)
215. 005528B  IF (TIMER) CALL SAUETI (INDIS, INAMES, KTIMES, TIMES, TFIRST)
216. 005529B  GO TO 600

C
C 501 CONTINUE
217. 005570B  IF (TIMER ) CALL SECOND(TFIRST)
218. 005571B  INIL = 1
219. 005572B  INDIS = 3
220. 005574B  CALL NATORD(NMXED, NDXUNK, INUNDX, MXNEQ)
221. 005575B  IF (TIMER) CALL SAUETI (ININD+15, INAMES, KTIMES, TIMES, TFIRST)
222. 005576B  GO TO 600

C
C 502 CONTINUE
223. 005584B  IF (TIMER) CALL SECOND(TFIRST)
224. 005594B  INSOL = 19
225. 005602B  DO 402 J = 1, MXNEQ
226. 005615B  402 UNKNOWN(J) = UNKNOWN(J)
598 INLOOP = 5
599 INLOOP = INLOOP - 1
600 IFC FATAL) GO TO 690
605 KSEG = KSEG + 1
606 MSEG = MODS(KSEG)
607 GO TO(500, 501, 502, 503, 504, 505, 506, 507, 508, 509)
650 CONTINUE
659 CALL SAVETI (45, INAMES, KTIMES, TIMES, TFIRST)
680 CONTINUE
689 CALL PLOT(0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
STOP
END
FUNCTION F(X,V)
RETURN
END

SUBROUTINE REAL CUALUS(1)
CUALUS( 2)
CUALUS( 3)
CUALUS( 4)
CUALUS( 5)
CUALUS( 6)
CUALUS( 7)
RETURN
END

REAL FUNCTION PDERHS(X, Y)
PDERHS = rex,'t)
RETURN
END

REAL FUNCTION BCONDCI, X, Y, BUALUS)
REAL BUALUS(4)
COMMON / INTEGS /
NUHBEQJNUMCOE.NRO.,.NCOL.NBAND.IROT(6).
ALEUEL.INITL.INDIS.INSO.,.MINPUT.MOUTPUT.MEMORV.ININD.
BNPDFX.D.NGRVDX.D.NGRIDZ,MXNEQ,MXNCOE.NROWD.NCOLD
IP = I + IROT(I)
GO TO (1, 2, 3.
1 BVAlUS(I)
2 BVAlUS(2)
3 BVAlUS(3)
4 BVAlUS(4)
999 BCOND = BUALUS(4)
100 CONTINUE
100 BCONDCI.
END

FUNCTION F(X,Y)
C
**** RIGHT SIDE OF THE PDE (DEPENDS ON U)
C
IX = X/HX + 0.001
IY = Y/HY + 0.001
IX = IX + (NGRIDX-2)*(IV-1)
IF ((IX .EQ. 0) .OR. (IX .EQ. (NGRIDX-1)) .OR.
A (IY .EQ. 0) .OR. (IY .EQ. (NGRIDY-1))) GO TO 50
UI = UNKNWN(I)
GO TO 100
GO TO 100
GO TO 100
GO TO 100
GO TO 100
GO TO 100
RETURN
END
1. FUNCTION TRUE(X,Y)
   C
   **** RETURN THE TRUE SOLUTION
2. TRUE = EXP(X*Y)
   RETURN
   END

1. SUBROUTINE TEST(I)
   C
   **** MONITOR CHANGE IN SOLUTION AT END OF EACH ITERATION
2. REAL DIFF(SO)
3. COMMON / UNKNWZ / UNKNW(IN)
4. COMMON / INTEGS / NUMEQ, MDUMB(N)
5. COMMON / OLDU(1) / UNKNW(K)
6. ITER = 5 - I
7. WRITE(6,1010) ITER
8. DIFMAX = 0.0
9. DO 10 J=1,NUMEQ
10. DIFF(J) = UNKNW(J) - UNKNW(J)
11. DIFMAX = AMAX1(DIFMAX,ABS(DIFF(J)))
12. 10 CONTINUE
13. WRITE(6,1020) DIFMAX, (J, DIFF(J), J=1,NUMEQ)
14. RETURN
   END

1. SUBROUTINE SETUP(U,N)
   C
   **** INITIALIZE SOLUTION AND INDEXING VECTORS
2. COMMON / MDUX22 / MDUX(IN)
3. COMMON / U(N)
4. DO 10 I=1,N
5. FI = I*0.001
6. U(I) = 1.0 + FI
7. MDUX(IN) = I
8. 10 CONTINUE
9. RETURN
10. END

---

ELLPACK 77 OUTPUT

+---------------------------------+
+ TABLE OF THE UNKNOWN ARRAY +
+ IN INTERNAL ORDER +
+---------------------------------+

<table>
<thead>
<tr>
<th>I</th>
<th>UNKNW(IN)</th>
<th>I</th>
<th>UNKNW(IN)</th>
<th>I</th>
<th>UNKNW(IN)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.045124E+00</td>
<td>7</td>
<td>1.293858E+00</td>
<td>12</td>
<td>1.645365E+00</td>
</tr>
<tr>
<td>2</td>
<td>1.091728E+00</td>
<td>8</td>
<td>1.367145E+00</td>
<td>13</td>
<td>1.730589E+00</td>
</tr>
<tr>
<td>3</td>
<td>1.138725E+00</td>
<td>9</td>
<td>1.396144E+00</td>
<td>14</td>
<td>1.397005E+00</td>
</tr>
<tr>
<td>4</td>
<td>1.183206E+00</td>
<td>10</td>
<td>1.293767E+00</td>
<td>15</td>
<td>1.645387E+00</td>
</tr>
<tr>
<td>5</td>
<td>1.226803E+00</td>
<td>11</td>
<td>1.464579E+00</td>
<td>16</td>
<td>1.325515E+00</td>
</tr>
<tr>
<td>6</td>
<td>1.201692E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table of the Indexing Arrays

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NDXEQ</th>
<th>NDXUNK</th>
<th>INUNDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
</tbody>
</table>

**MAX CHANGE IN SOLUTION**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NDXEQ</th>
<th>NDXUNK</th>
<th>INUNDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.05E-03</td>
<td>2</td>
<td>-9.09E-03</td>
</tr>
<tr>
<td>5</td>
<td>-9.05E-03</td>
<td>6</td>
<td>-1.79E-02</td>
</tr>
<tr>
<td>9</td>
<td>-1.19E-02</td>
<td>10</td>
<td>-2.42E-02</td>
</tr>
<tr>
<td>13</td>
<td>-1.02E-02</td>
<td>14</td>
<td>-2.12E-02</td>
</tr>
</tbody>
</table>

**MAX CHANGE IN SOLUTION**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NDXEQ</th>
<th>NDXUNK</th>
<th>INUNDX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.69E-04</td>
<td>2</td>
<td>6.37E-04</td>
</tr>
<tr>
<td>5</td>
<td>6.96E-04</td>
<td>6</td>
<td>1.36E-03</td>
</tr>
<tr>
<td>9</td>
<td>8.82E-04</td>
<td>10</td>
<td>1.76E-03</td>
</tr>
<tr>
<td>13</td>
<td>7.08E-04</td>
<td>14</td>
<td>1.42E-03</td>
</tr>
</tbody>
</table>
**ELLPACK 77 OUTPUT**

```
+ MAX( ABSCERROR ) ON 6 X 6 GRID = 4.3465084E-06 +
```

```
MAX CHANGE IN SOLUTION = 1.53292E-04 DIFFERENCES ARE

1  -2.40E-05  2  -4.79E-05  3  -6.09E-05  4  -4.74E-05
5  -4.79E-05  6  -9.28E-05  7  -1.19E-04  8  -9.31E-05
9  -6.09E-05 10  -1.19E-04 11  -1.52E-04 12  -1.22E-04
13 -4.73E-05 14  -9.30E-05 15  -1.22E-04 16  -8.78E-05
```

---

**ELLPACK 77 OUTPUT**

```
+ MAX( ABSCERROR ) ON 6 X 6 GRID = 1.1451043E-05 +
```

```
MAX CHANGE IN SOLUTION = 9.92862E-06 DIFFERENCES ARE

1   1.64E-06 2   3.17E-06 3   4.02E-06 4   3.12E-06
5   3.17E-06 6   6.12E-06 7   7.76E-06 8   6.05E-06
9   4.02E-06 10  7.76E-06 11  9.93E-06 12  7.76E-06
13  3.11E-06 14  6.05E-06 15  7.76E-06 16  6.05E-06
```

---

**ELLPACK 77 OUTPUT**

```
+ MAX( ABSCERROR ) ON 6 X 6 GRID = 1.0610114E-05 +
```

```
MAX CHANGE IN SOLUTION = 6.40925E-07 DIFFERENCES ARE

1  -1.07E-07 2  -2.03E-07 3  -2.62E-07 4  -2.03E-07
5  -2.03E-07 6  -3.91E-07 7  -5.05E-07 8  -3.91E-07
9  -2.62E-07 10  -5.05E-07 11  -6.41E-07 12  -4.99E-07
13 -2.03E-07 14  -3.91E-07 15  -4.99E-07 16  -3.87E-07
```
### EXECUTION TIMES

<table>
<thead>
<tr>
<th>MODULE NAME</th>
<th>SECONDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.067</td>
</tr>
<tr>
<td>2</td>
<td>0.017</td>
</tr>
<tr>
<td>3</td>
<td>0.028</td>
</tr>
<tr>
<td>4</td>
<td>2.905</td>
</tr>
<tr>
<td>5</td>
<td>0.114</td>
</tr>
<tr>
<td>6</td>
<td>0.067</td>
</tr>
<tr>
<td>7</td>
<td>0.015</td>
</tr>
<tr>
<td>8</td>
<td>0.025</td>
</tr>
<tr>
<td>9</td>
<td>0.118</td>
</tr>
<tr>
<td>10</td>
<td>0.021</td>
</tr>
<tr>
<td>11</td>
<td>0.085</td>
</tr>
<tr>
<td>12</td>
<td>0.015</td>
</tr>
<tr>
<td>13</td>
<td>0.027</td>
</tr>
<tr>
<td>14</td>
<td>0.123</td>
</tr>
<tr>
<td>15</td>
<td>0.018</td>
</tr>
<tr>
<td>16</td>
<td>0.068</td>
</tr>
<tr>
<td>17</td>
<td>0.001</td>
</tr>
<tr>
<td>18</td>
<td>0.034</td>
</tr>
<tr>
<td>19</td>
<td>0.124</td>
</tr>
<tr>
<td>20</td>
<td>0.020</td>
</tr>
<tr>
<td>21</td>
<td>0.036</td>
</tr>
<tr>
<td>22</td>
<td>0.016</td>
</tr>
<tr>
<td>23</td>
<td>0.029</td>
</tr>
<tr>
<td>24</td>
<td>0.138</td>
</tr>
<tr>
<td>25</td>
<td>0.019</td>
</tr>
<tr>
<td>26</td>
<td>0.068</td>
</tr>
<tr>
<td>27</td>
<td>0.001</td>
</tr>
<tr>
<td>28</td>
<td>0.017</td>
</tr>
<tr>
<td>29</td>
<td>0.023</td>
</tr>
<tr>
<td>30</td>
<td>0.135</td>
</tr>
<tr>
<td>31</td>
<td>0.019</td>
</tr>
<tr>
<td>32</td>
<td>0.019</td>
</tr>
<tr>
<td>33</td>
<td>4.428</td>
</tr>
</tbody>
</table>

**TOTAL EXECUTION**
APPENDIX THREE - ELLPACK 77 MODULES

All of the modules currently available in ELLPACK 77 are described briefly in this appendix. The 31 modules are listed below; the numbers are not used in ELLPACK itself, they provide a simple designation of the programs.

<table>
<thead>
<tr>
<th>Discretization</th>
<th>Indexing</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. 5-POINT STAR</td>
<td>14. NATURAL</td>
<td>20. LINPACK BAND</td>
</tr>
<tr>
<td>2. 7-POINT 3D</td>
<td>15. RED-BLACK</td>
<td>21. LINPACK SPD BAND</td>
</tr>
<tr>
<td>3. PC-CI COLLOCATION</td>
<td>16. YALE MIN DEG</td>
<td>22. SPARSK GL-PIVOTING</td>
</tr>
<tr>
<td>4. P3-CI GALERKIN</td>
<td>17. YALE RCM</td>
<td>23. SOR</td>
</tr>
<tr>
<td>5. HODIE-HELMHOLTZ</td>
<td>32. NESTED DISSECTION</td>
<td>24. JACOBI SI</td>
</tr>
<tr>
<td>6. HODIE-ACP</td>
<td></td>
<td>25. JACOBI CG</td>
</tr>
<tr>
<td>8. HODIE 27-POINT 3D</td>
<td></td>
<td>26. SYMMETRIC SOR SI</td>
</tr>
<tr>
<td>9. FFT 9-POINT</td>
<td></td>
<td>27. SYMMETRIC SOR CG</td>
</tr>
<tr>
<td>10. 2 DEFEP</td>
<td></td>
<td>28. REDUCED SYSTEM CI</td>
</tr>
<tr>
<td>11. MARCHING ALGORITHM</td>
<td></td>
<td>29. REDUCED SYSTEM CG</td>
</tr>
<tr>
<td>12. DYAKANOV-CG</td>
<td></td>
<td>30. YALE SPARSE</td>
</tr>
<tr>
<td>13. DYAKANOV-CG</td>
<td></td>
<td>31. YALE ENVELOPE</td>
</tr>
</tbody>
</table>

Modules 18, 19 and 23-24 have been retired, 7 is reserved for HODIE-ACP.

Figure A4 illustrates a set of module sequences which are legal in ELLPACK 77. Some paths have been left out of this table, however. For example, the sequence 3-16-22 will work, but the sequence 3-16-30 will not (since YALE SPARSE does not do partial pivoting). Thus the table indicates that module 16 cannot in general follow module 3. The user should consult the following module descriptions for more details about compatibility.

<table>
<thead>
<tr>
<th>MODULE</th>
<th>MAY BE FOLLOWED BY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14, 15, 16, 17</td>
</tr>
<tr>
<td>2</td>
<td>14, 15, 16, 17</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>14, 16, 17</td>
</tr>
<tr>
<td>6</td>
<td>14, 16, 17</td>
</tr>
<tr>
<td>7</td>
<td>14, 16, 17</td>
</tr>
<tr>
<td>8-13</td>
<td>None</td>
</tr>
<tr>
<td>14</td>
<td>20, 21, 22, 30, 31, 33, 34, 35, 36, 37</td>
</tr>
<tr>
<td>15</td>
<td>30, 31, 38, 39</td>
</tr>
<tr>
<td>16</td>
<td>22, 30, 31</td>
</tr>
<tr>
<td>17</td>
<td>22, 30, 31</td>
</tr>
<tr>
<td>32</td>
<td>22, 30, 31</td>
</tr>
</tbody>
</table>

Figure A4. Module Compatibility Table
MODULE NAME: 5-POINT STAR

AUTHOR/DATE: Ron Boisvert, 4/20/77

INITIAL/FINAL INTERFACES: EQUATION FORMATION - EQUATION INDEXING

MODULE FUNCTION: Discretizes a General Operator with Dirichlet, Neuman or Mixed Boundary Conditions.

RESTRICTIONS ON USE:

Two dimensions

No UXY term

Uniform grid of size at least 3x3

METHOD DESCRIPTION: Usual 5-point star discretization, derivative boundary conditions approximated by one-sided $O(h^2)$ differences, or, for self-adjoint operators, by a special symmetric formula for normal derivatives.

PARAMETERS: None

KEYWORDS THAT AFFECT MODULE: Constant Coefficients, Self-adjoint

STORAGE AND TIMING ESTIMATES: Approximately $NGRIDX \times NGRIDY$ equations are generated with 6 coefficients per equation. A workspace of $42 + NGRIDX*NGRIDY$ single precision words is used.
MODULE NAME: P3-C1 GALERKIN

AUTHOR/DATE: E.N. Houstis 1/20/78

INITIAL/FINAL INTERFACE: EQUATION FORMATION - EQUATION INDEXING

MODULE: Discretizes a self-adjoint elliptic operator with homogeneous Dirichlet or Neumann Conditions

RESTRICTION ON USE: Two dimensions
Homogeneous boundary conditions

METHOD DESCRIPTION: Galerkin based on bicubic Hermite elements
Elements satisfy exactly Dirichlet or Neumann boundary conditions

PARAMETERS: None

KEYWORDS THAT AFFECT MODULE: DIRICHLET, NEUMANN

STORAGE AND TIMING ESTIMATES:

4 + (NGRIDX-1) * (NGRIDY-1) equations are generated with at most 36 nonzero coefficients per equation

Since the system of equations is symmetric only the upper triangular half is stored. A workspace of size
720 + (NGRIDX-1)*(NGRIDY-1)*(4*NGRIDY+4)
is required.

NOTE: The April, 1980 version has the following RESTRICTION on the grid size: The largest grid is 10 x 10.
MODULE NAME: P3-C1 COLLOCATION

AUTHOR/DATE: Elias Houstis, 5/20/77

INITIAL/FINAL INTERFACE: EQUATION FORMATION-EQUATION INDEXING

MODULE FUNCTION: Discretizes a General Operator with Dirichlet, Neumann or Mixed Boundary Conditions.

RESTRICTIONS ON USE:

Two dimensions

Grid of size at least 3x3

METHOD DESCRIPTION: Collocation based on bicubic Hermite elements.

Interpolates the nonhomogeneous boundary conditions.

Elements satisfy exactly Dirichlet or Neumann homogeneous boundary conditions.

PARAMETERS: None

KEYWORDS THAT AFFECT MODULE: homogeneous (Boundary Conditions)

STORAGE AND TIMING ESTIMATES:

Nonhomogeneous boundary conditions: \( 4 \times \text{NGRIDX} \times \text{NGRIDY} \)

equations are generated with 16 coefficients per equation

Homogeneous boundary conditions: \( 4 \times (\text{NGRIDX}-1) \times (\text{NGRIDY}-1) \)

equations are generated with 16 coefficients per equation

Note: The April, 1980 version has the following RESTRICTION on the grid size:

A. As distributed by INSL, largest grid is 14 x 14
B. As run on Purdue CDC6500, largest grid is 10 x 10
C. As run on Purdue Vax, largest grid is 23 x 23
MODULE NAME: HODIE-HELMHOLTZ

AUTHOR/DATE: Ron Boisvert, September 7, 1978

INITIAL/FINAL INTERFACES: EQUATION FORMATION-EQUATION INDEXING

MODULE FUNCTION: Discretizes the Dirichlet problem for an operator of the form \( u_{xx} + u_{yy} + f(x,y)u = g(x,y) \). When applicable, this module should be used instead of HODIE-ACF.

RESTRICTIONS ON USE: Two dimensions, uniform grid of size 3 \times 3, h_x=h_y.

METHOD must be 41 or 42 if \( f(x,y) \) is not a constant (see PARAMETERS).

METHOD DESCRIPTION: A 4th or 6th order 9-point finite difference approximation (HODIE) of the form

\[
8 \sum_{i=0}^{8} a_i u(x_i, y_i) = h^2 \sum_j b_j g(x_j, y_j)
\]

where the points \((x_i, y_i)\) are the usual 9-point star points and the points \((x_j, y_j)\) are used to get high order. Sixth order methods are attained using superpositions of 4th order methods. See Boisvert, R. F., The effect of accuracy of the placement of auxiliary points in the HODIE method for the Helmholtz problem, Purdue University, C.S. Department, CSD-TR 266, June 1978.

PARAMETERS: METHOD determines choice of auxiliary points.

<table>
<thead>
<tr>
<th>METHOD</th>
<th>ORDER</th>
<th>NUMBER AND LOCATION OF AUXILIARY POINTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>4th</td>
<td>4 at midpoints of subrectangles</td>
</tr>
<tr>
<td>41</td>
<td>4th</td>
<td>4 at side grid points</td>
</tr>
<tr>
<td>42</td>
<td>4th</td>
<td>4 at corner grid points</td>
</tr>
<tr>
<td>43</td>
<td>4th</td>
<td>4 at optimal points (see reference)</td>
</tr>
<tr>
<td>60</td>
<td>6th</td>
<td>12, superposition of 40, 41, 42 above</td>
</tr>
<tr>
<td>61</td>
<td>6th</td>
<td>8, same as 62 with ( n=5/27 )</td>
</tr>
<tr>
<td>62</td>
<td>6th</td>
<td>8, choose from a one-parameter family by specifying ETA=n, where n[0,1]</td>
</tr>
</tbody>
</table>

KEY-WORDS THAT AFFECT MODULE: CONSTANT COEFFICIENTS

STORAGE AND TIMING ESTIMATES: A workspace of size 50 + NGIDX*NGRIDEY is required.
MODULE NAME: HODIE-ACF

AUTHOR/DATE: Ron Boisvert, 3/27/78

INITIAL/FINAL INTERFACES: EQUATION FORMATION-EQUATION INDEXING

MODULE FUNCTION: Discretizes the Dirichlet problem for an operator of
the form \( a(x,y)u_{xx} + c(x,y)u_{yy} + f(x,y)u = g(x,y) \)

RESTRICTIONS ON USE: Two dimensions, uniform grid of size at least 3x3,
no \( u_{xy}, u_x \) or \( u_y \) terms. Must have \( a(x,y) = c(x,y) \) for 6th order.

METHOD DESCRIPTION: A 4th or 6th order 9-point difference approximation
via identity expansion (HODIE) of the form
\[
\sum_{i=0}^{8} a_i u(x_i,y_i) = h_x h_y \sum_{j} b_j g(x_j,y_j)
\]
where the points \((x_i,y_i)\) are on a grid of the form \( \square \) and the auxiliary
points \((x_j,y_j)\) lie in the same region. The number of auxiliary points
depends upon the operator and the order of the method; it may vary from
5 to 28. See Lynch, R.E. and Rice, J.R., "High Accuracy Approximation
to Solutions of Elliptic Partial Differential Equations", Purdue Univ.,

PARAMETERS: METHOD = n, where n is the order of the method; n may be 4 or 6.

In the case of 4th order methods for constant coefficient operators ORDER=41
or 42 may be specified to yield slightly different auxiliary points. (Defaults to

KEYWORDS THAT AFFECT MODULE: CONSTANT COEFFICIENTS; setting LEVEL=3 yields a
listing of auxiliary points and a and b coefficients.

STORAGE AND TIMING ESTIMATES: A workspace of size 4025+NGRIDX*NGRIDY is required
MODULE NAME: HODIE-ACDEF

AUTHOR/DATE: Robert E. Lynch, *** (NOT YET AVAILABLE)

INITIAL/FINAL INTERFACES: EQUATION FORMATION-EQUATION INDEXING

MODULE FUNCTION: Discretized the Dirichlet problem for an equation of the form

\[ Au_{xx} + Cu_{yy} + Du_x + Eu_y + Fu = G, \]

with \( A, C, D, E, F, G \) functions of \( x \) and \( y \).

RESTRICTIONS ON USE: Two dimensions, uniform grid of size at least 3-by-3.

METHOD DESCRIPTION: The discrete equations is a HODIE equation of the form

\[ (1/Ax^2) \sum_{i=0}^B \alpha_i u(x_i, y_i) = \sum_{j=1}^J \beta_j G(n_j, \xi_j) \]

with the coefficients determined so that the approximation is exact on polynomials of degree at least 5; this gives \( O(h^4) \) estimation of \( u \in C^6 \).


PARAMETERS: AA, CC, DD, EE, FF, GG, WARN.

\( DD=1, DD=0, \) or \( DD=-1, \) denotes, respectively, that the coefficient \( D \) depends on \( x \) and \( y \), is zero, or is constant; similarly for \( AA, CC, EE, FF \) and \( GG; AA=0 \) or \( CC=0 \) results in a fatal error. The default is \( 1 \), except when the keywords CONSTANT COEFFICIENTS appear, then the default is \(-1 \) for \( AA, CC, DD, EE, FF \) and \( 1 \) for \( GG \). \( WARN=0, WARN=1, \) or \( WARN=2, \) denotes, respectively, that no warning messages are to be printed, that a message is to be printed if the system of equations which determines the \( \beta 's \) is ill-conditioned (usually means that the grid-spacing is too large), or the messages for \( 1 \) and in additions messages if the sign of \( F \) is the same as the sign of \( A \).

KEYWORDS THAT AFFECT MODULE: CONSTANT COEFFICIENTS.

STORAGE AND TIMING ESTIMATES: A workspace of about \( 700 + 17 \times (NGRIDX \times NGRIDY) \) is used.

REMARKS: When NATRUAL ORDER is used with a band-solver, arrange coordinate system so that \( NGRIDY \) is \( \leq \) \( NGRIDX \) to reduce band-width.
MODULE NAME: HODIE 27-POINT 3D

AUTHOR/DATE: R. E. Lynch, 2/15/77

(Modified for ELLPACK use, Ron Boisvert, 6/1/77)

INITIAL/FINAL INTERFACES: EQUATION FORMATION - OUTPUT

MODULE FUNCTION: Discretizes the Dirichlet problem for the Poisson equation on a cube and solves the resulting equations

RESTRICTIONS ON USE:

Three dimensions
Poisson equation
Dirichlet boundary conditions
Cubical domain
Uniform grid with \( HX = HY = HZ \) and at least 3x3x3


PARAMETERS: None

KEYWORDS THAT AFFECT MODULE: None

STORAGE AND TIMING ESTIMATES: A workspace of size \( 2N^3 + 4N - 2 \) is required.

Number of unknowns is \( (N-2)^3 \)
MODULE NAME: FFT 9-POINT

AUTHOR/DATE: E.N. Houstis and T.S. Papatheodorou 10/25/77

INITIAL/FINAL INTERFACE: EQUATION FORMATION - EQUATION SOLUTION

MODULE FUNCTION: Uses a 2-nd, 4-th or 6-th order 9-point difference formula and fast Fourier transform for the numerical solution of the elliptic equation with constant coefficients

\[ au_{xx} + bu_{yy} + cu = f \]

on a rectangle subject to Dirichlet boundary conditions


RESTRICTIONS ON USE: Must have NGRIDX = 2 ** I+1, NGRIDY = 2 ** J+1 with I, J > 3. 6-th order applies only to the problem a=b=1 with Hx = Hy.

METHOD DESCRIPTION: The algorithm consists of the following components

(i) Grid: uniform NGRIDX = 2^I + 1, NGRIDY = 2^J + 1

(ii) Discretization operator: 4th and 6th 9-point finite difference schemes

(iii) Equation solver:

(a) Odd/even reduction
(b) Fourier Analysis on even lines.
(c) Recursive Cyclic Reduction
(d) Fourier synthesis on the even lines
(e) Solution on the odd lines

PARAMETERS: IORDER = n, where n is the order of the method (can be 2, 4 or 6)

KEYWORDS THAT AFFECT MODULE: None

STORAGE AND TIMING ESTIMATES: Approximate working space

Order = 6  7 + 10*max(NGRIDX,NGRIDY) + 2 * NGRIDX * NGRIDY

Order = 4  7 + 10*max(NGRIDX,NGRIDY) + NGRIDX * NGRIDY

Note: In the April, 1980 version further limitations on the domain are:

1. IORDER = 2: Must have HX = HY

2. IORDER = 4 and 6: Must have lower left corner of domain = (0,0)
MODULE NAME: 2DEPEP
AUTHOR/DATE: G. Sewell, 9/13/78
INITIAL/FINAL INTERFACE: EQUATION FORMATION-OUTPUT

RESTRICTIONS: Two-dimensions, self-adjoint, uniform grid (output at grid points). Not usable for ELLPACK 77, can use ELLPACK 78 for rectangular domains.

MODULE DESCRIPTION: Galerkin's method with 6-node triangular elements, user-controlled grading of the triangular mesh, and the frontal method to organize out-of-core storage of the matrix when necessary.

PARAMETERS: NTRI = number of triangles desired

MEM = 46*NTRI + NDM where NDM = 35*NTRI 1f external storage used

= 15*NTRI**1.5 otherwise

FORTRAN: The Fortran Function D3EST(X,Y) must be provided.

Ideally D3EST should be an estimate of max_{1+J=3} \left| \frac{\partial^3 u}{\partial x \partial y^2} \right|.

In any case, the program grades the triangulation so that it is most refined where D3EST is largest, and attempts to distribute D3EST, \( H_j^{**3} \) uniformly. (\( H_j \) = diameter of triangle \( J \)).

This module is a small, specialized subset of the program TWODEPEP, a commercial product of IMSL, Inc. For further details see G. Sewell, A finite element program with automatic user controlled mesh grading, in Advances in Computer Methods for Partial Differential Equations III, (R. Vishnevetsky, ed), Rutgers Univ., New Brunswick, N.J.
MODULE NAME: MARCHING ALGORITHM
AUTHOR/DATE: R. E. Bank, August 1978
INITIAL/FINAL INTERFACES: EQUATION FORMATION/SOLUTION
MODULE FUNCTION: Solves separable, self-adjoint problems with general boundary conditions,

RESTRICTIONS ON USE: Uniform grid, 2-dimensional, self-adjoint, separable
NGRIDX ≥ 3, 4 or 5 for 0, 1 or 2, respectively, Dirichlet boundary conditions in x-direction. Similarly for NGRIDY.

METHOD DESCRIPTION: A symmetric 5-point finite difference discretization is used. The linear equations are solved using the generalized marching algorithm. A least-squares solution is computed for singular but semi-definite problems. For further details see R. Bank, Algorithm 527: A Fortran implementation of the generalized marching algorithm, ACM Trans. Math. Software, 4, (1978) pp. 165-176.

PARAMETERS: KGMA: marching parameter, integer, default = 2.

KEY-WORDS THAT AFFECT MODULE: UNIFORM, 2 DIMENSIONS, SELF-ADJOINT, CHOSEN TERMS

STORAGE AND TIMING ESTIMATES: Operation count is the order of NGRIDX*NGRIDY*log(NGRIDY/KGMA).
MODULE NAME: DYAKANOV-CG

AUTHOR/DATE: R. E. Bank, August 1978

INITIAL/FINAL INTERFACES: EQUATION FORMATION/SOLUTION

MODULE FUNCTION: Solves self-adjoint problems with general boundary conditions

RESTRICTIONS ON USE: Uniform grid, 2-dimensional, self adjoint; NGRIDX >
3, 4 or 5 for 0, 1 or 2, respectively, Dirichlet boundary conditions
in the x-direction. Similarly for NGRIDY.

METHOD DESCRIPTION: A symmetric 5-point discretization is used. The
equations are solved using a preconditioned conjugate gradient
iteration. A scaled, separable approximation of the nonseparable
matrix is used as the preconditioning matrix. The generalized
marching algorithm is used to solve the separable problems. The
initial guess is zero. For further details see R. Bank, Algorithm
527: A Fortran implementation of the generalized marching algorithm, ACM

PARAMETERS: ITIMAX: integer, maximum number of CG iterations.
DEMAND: Real, accept solution when the estimated error in the linear
equations problem is reduced by \(10^{-\text{DEMAND}}\).

KEY-WORDS THAT AFFECT MODULE:
UNIFORM, TWO DIMENSIONS, SELF-ADJOINT, CROSS TERMS

STORAGE AND TIMING ESTIMATES: Operation count is the order of NGRIDX*
NGRIDY*\log(NGRIDY)*\text{ITNUM} where \text{ITNUM} = number of iterations;
Storage = 7*NGRIDX*NGRIDY + NGRIDY*\log_2(NGRIDY) + lower-order terms
MODULE NAME: DYAKANOV-CG 4

AUTHOR/DAT: R. E. Bank, August 1978

INITIAL/FINAL INTERFACES: EQUATION FORMATION/SOLUTION

MODULE FUNCTION: Solves self-adjoint problems with general boundary conditions

RESTRICIONS ON USE: Uniform grid, 2-dimensional, self-adjoint; NGIDX and NGIDY must odd integers; NGIDX \geq 5, 7 or 9 for 0, 1 or 2, respectively, Dirichlet boundary conditions in the x-direction. Similarly for NGIDY.

METHOD DESCRIPTION: A symmetric 5-point discretization is used. Fourth-order accuracy is achieved by solving the problem on a coarse grid, \((\text{NGIDX+1})/2 \times (\text{NGIDY+1})/2\), and a fine grid, \((\text{NGIDX}) \times (\text{NGIDY})\), and using Richardson extrapolation at grid points in the fine mesh. The equations are solved using a preconditioned conjugate gradient iteration. A scaled, separable approximation of the nonseparable matrix is used as the preconditioning matrix. The generalized marching algorithm is used to solve the separable problems. For further details see R. Bank, Algorithm 527: A Fortran implementation of the generalized marching algorithm, ACM Trans. Math. Software, 4, (1978) pp. 165-176.

PARAMETERS: ITMAX: integer, maximum number of CG iterations in each grid.

DEMAND: Real, accept solution when estimated error in the linear equations problem is reduced by 10^{-DEMAND}.

KEY-WORDS THAT AFFECT MODULE: UNIFORM, DIM2, SELFAD, CROSSST

STORAGE AND TIMING ESTIMATES: Operation count is the order of \(\text{NGIDX} \times \text{NGIDY} \times \log(\text{NGIDY}) \times \text{ITNUM}\) where \text{ITNUM} = number of iterations.

Storage = 8*\text{NGIDX}*\text{NGIDY} + \text{NGIDY}*\log(\text{NGIDY}) + lower-order terms.
MODULE NAME: 7-POINT 3D

AUTHOR/DATE: Roger Grimes, June 1978

INITIAL/FINAL INTERFACES: EQUATION FORMATION-EQUATION INDEXING

MODULE FUNCTION: Discretizes a general self-adjoint elliptic operator with Dirichlet, Neumann or mixed boundary conditions on a rectangular box.

RESTRICTIONS ON USE: Three dimensions, self-adjoint equation, rectangular domain size at least 3 x 3 x 3, no UXY, UYZ, UX, UY or UZ terms.

METHOD DESCRIPTION: Discretizes the general self-adjoint problem

\[(a_{xx}u_x + (b_{yy}u_y + (c_{zz}u_z)) + f = g\]

using 7-point symmetric differences on a rectangular mesh. A symmetric positive definite matrix is generated for all boundary conditions. Non-uniform meshes are allowed.

PARAMETERS: None

KEY-WORDS THAT AFFECT MODULE: SELF-ADJOINT

STORAGE AND TIMING ESTIMATES: COEF and IDCOEF dimensioned "number of interior points plus number of non-Dirichlet boundary points" x 8.

NGRIDX*NGRIDY*NGRIDZ x 8
MODULE NAME: RED-BLACK
AUTHOR/DATE: R. Grimes, June 1978
INITIAL/FINAL INTERFACES: EQUATION INDEXING-EQUATION SOLUTION
MODULE FUNCTION: Indexes equations and unknowns in red-black ordering.
RESTRICTIONS ON USE: Matrix obtained from 5-POINT STAR or 7-POINT 3D discretization

METHOD DESCRIPTION: Using ICCOEF, all points are labeled either red or black such that red points are adjacent to black points only and vice-versa. The red-black index array NDXEQ is ordered such that all red points are numbered first followed by all black points. If a red-black ordering is not possible, the module prints an error message and returns.

PARAMETERS: None

KEY-WORDS THAT AFFECT MODULE: None

STORAGE AND TIMING ESTIMATES:
MODULE NAME: NATURAL
AUTHOR/DATE: Ron Boisvert, 6/1/77
INITIAL/FINAL INTERFACES: EQUATION INDEXING - EQUATION SOLUTION
MODULE FUNCTION: Indexed equations using the natural ordering
RESTRICTIONS ON USE: None
METHOD DESCRIPTION: Equations are indexed in the order generated in
the discretization module.
PARAMETERS: None
KEYWORDS THAT AFFECT MODULE: None
STORAGE AND TIMING ESTIMATES: No workspace storage is required.
MODULE NAME: YALE MIN DEG
AUTHOR/DATE: Andrew H. Sherman, 9/15/78
INITIAL/FINAL INTERFACES: EQUATION INDEXING/EQUATION SOLUTION
MODULE FUNCTION: Computes an indexing
RESTRICTIONS ON USE: None
METHOD DESCRIPTION: Applies minimum degree algorithm to $A + A^T$
   ($A$ is the coefficient matrix)
PARAMETERS: NSP = amount of working storage (see note for YALE SPARSE module)
KEY-WORDS THAT AFFECT MODULE: None
STORAGE AND TIMING ESTIMATES: Storage: NSP is amount of working storage
   Timing: variable
Module Name: NESTED DISSECTION

Author/Date: Andrew H. Sherman, 05/23/79

Initial/Final Interfaces: Equation indexing/equation solution

Module Function: Computes an indexing for grid-based equations and unknowns

Restrictions on Use: There must be some association between the equations/unknowns and grid points. Region must be in two dimensions.

Method Description: The region is assumed to be embedded in an X,Y grid.

A nested dissection ordering is performed on the smallest square in which the X,Y grid can be embedded, and the actual points in the region are numbered in the same relative ordering as in this nested dissection ordering. The unknowns of the discretized equations are then numbered in the same relative order as the grid points with which they are associated, and the equations are given the same order as the unknowns. If several unknowns are associated with the same grid point, they are numbered consecutively in increasing order by original unknown number. For further details see S.C. Eisenstat, M.G. Gursky, M.H. Schultz and A.H. Sherman, Yale Sparse Matrix Package, I: The Symmetric Codes, Report 112. II: The Non-symmetric Codes, Report 114, Computer Science, Yale University, 1977

Parameters: NTYPE = 5 for Five-Point nested dissection.

NDTYPE, N.E. 5 for Nine-Point nested dissection.

Keywords that affect Module: DDIM2 must be true.

Storage and Timing Estimates: Workspace required = 2*MAX(NGRIDX,NGRIDY)**2
Module Name: YAEL RCM

Author/Date: Andrew H. Sherman, 05/23/79

Initial/Final Interfaces: Equation indexing/equation solution

Module Function: Computes an indexing for use with envelope or band solvers.

Restrictions on Use: Matrix must have symmetric structure


Parameters: None

Keywords that Affect Module: None

Storage and Timing Estimates: No workspace storage is required
MODULE NAME: LINPACK BAND
AUTHOR/DATE: Cleve Moler, 3/11/78
INITIAL/FINAL INTERFACE: EQUATION SOLUTION - OUTPUT
MODULE FUNCTION: Solves a real band system of linear equations
RESTRICTIONS ON USE: None
METHOD DESCRIPTION: Reformats the linear system into band matrix storage
(diagonals stored in rows) and uses Gauss elimination with partial pivoting to solve the band system. An LU factorization is used.
The solution phase of this module is a product of the LINPACK project supported by the National Science Foundation. See J.J. Dongarra et al, LINPACK Users Guide, SIAM, Philadelphia, Pa. (1979)

PARAMETERS: None
KEY-WORDS THAT AFFECT MODULE: None
STORAGE AND TIMING ESTIMATES: For a system of N equations with ML diagonals below the main diagonal and MU diagonals above the main diagonal, the generated band matrix has size \((2*ML+MU+1) \times N\). In addition, a workspace of size \(N\) is required.
MODULE NAME: LINPACK SPD BAND
AUTHOR/DATE: Cleve Moler, 3/11/78
INITIAL/FINAL INTERFACE: EQUATION SOLUTION - OUTPUT
MODULE FUNCTION: Solves a real symmetric positive definite band system
of linear equations
RESTRICTIONS ON USE: None
METOD DESCRIPTION: Reformats the linear system into symmetric band
matrix storage (diagonals stored in rows) and used Cholesky
decomposition to solve the symmetric positive definite band system.
The solution phase of this module is a product of the LINPACK
project supported by the National Science Foundation. See J.J. Dongarra
PARAMETERS: None
KEY-WORDS THAT AFFECT MODULES: None
STORAGE AND TIMING ESTIMATES: For a system of $N$ equations with $M$
diagonals above the main diagonal, a band matrix of size $(M+1) \times N$
is generated. No extra workspace is required.
COMPATIBLE WITH ELLPACK VERSIONS: September 1978
MODULE NAME: SPARSE GE-PIVOTING

AUTHOR/DATE: Andrew Sherman

(Modified for ELLPACK USE, Ron Boisvert, 7/1/78)

INITIAL/FINAL INTERFACES: EQUATION SOLUTION - OUTPUT

MODULE FUNCTION: Converts equations to sparse storage mode and solves them

RESTRICTIONS ON USE: None (See note below)

METHOD DESCRIPTION: Sparse Gauss Elimination with column interchanges.


METHOD DESCRIPTION: Sparse Gauss Elimination with column interchanges.

PARAMETERS:

MAXNZ = maximum number of off-diagonal non-zero entries in the upper-triangular factor of the matrix

KEYWORDS THAT AFFECT MODULE: TIME

STORAGE AND TIMING ESTIMATES: The matrix is stored in a linear array of size number of equations * max number of coefficients per equation.

A workspace of size 3*MAXNZ + 6* number of equations + 3 is required.

NOTE: A module interface incompatibility makes this module incompatible with the indexing module RED-BLACK in the April, 1980 version of ELLPACK.
MODULE NAME:  JACOBI CG
JACOBI SI
SOR
SYMMETRIC SOR CG
SYMMETRIC SOR SI
REDUCED SYSTEM CG
REDUCED SYSTEM SI


These modules were contributed by the ITPACK project of the Center for Numerical Analysis at the University of Texas at Austin.

INITIAL/FINAL INTERFACES:  EQUATION SOLUTION-OUTPUT

MODULE FUNCTION:  These ITPACK modules solve a system of symmetric linear equations in sparse symmetric storage mode by adaptive iterative algorithms with automatic parameter determination and stopping tests.

RESTRICTIONS ON USE:  Symmetric positive definite systems of linear equations, red-black ordering for REDUCED SYSTEM modules.


PARAMETERS:  The following optional parameters allow a certain degree of control over the iterative algorithms.

<table>
<thead>
<tr>
<th></th>
<th>ITIMAX</th>
<th>ZETA</th>
<th>OME</th>
<th>SME</th>
<th>OMEGA</th>
<th>SPECR</th>
<th>BETA</th>
<th>IADAPT</th>
<th>ICASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACOBI CG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>JACOBI SI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SOR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SYMMETRIC SOR CG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SYMMETRIC SOR SI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>REDUCED SYSTEM CG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>REDUCED SYSTEM SI</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Parameter</td>
<td>Meaning or Usage</td>
<td>Default</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITMAX</td>
<td>Maximum number of iterations allowed</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ZETA</td>
<td>Tolerance level in stopping test</td>
<td>10^-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CME</td>
<td>Initial estimate of largest eigenvalue of Jacobi matrix</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SME</td>
<td>Initial estimate of smallest eigenvalue of Jacobi matrix</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OMEGA</td>
<td>Overrelaxation parameter for SOR and SSOR methods</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPECR</td>
<td>Initial estimate of spectral radius of SSOR matrix</td>
<td>0.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BETA</td>
<td>Initial estimate of spectral radius of LU matrix in SSOR method</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IADAPT</td>
<td>Switch indicating whether adaptive procedure should be used</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 0 - fixed values for iteration parameters CME, SME, OMEGA, SPECR, BETA
- 1 - adaptive procedures used for values of CME, SME, OMEGA, SPECR, BETA
- 2 - (SSOR method only) fixed values for CME, OMEGA and adaptive value for SPECR
- 3 - (SSOR method only) fixed value for BETA and adaptive values for CME, OMEGA, SPECR

<table>
<thead>
<tr>
<th>ICASE</th>
<th>Switch indicating which case of the adaptive procedure to use</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Abs(SME) .GE. CME, SME fixed for JSI</td>
</tr>
<tr>
<td>2</td>
<td>Abs(SME) .LT. CME, SME set to -CME</td>
</tr>
</tbody>
</table>

**KEY-WORDS THAT AFFECT MODULE:** INDEX RED-BLACK for Reduced System CG and SI

**STORAGE AND TIMING ESTIMATES:**

<table>
<thead>
<tr>
<th>Method</th>
<th>Workspace size for N unknowns</th>
</tr>
</thead>
<tbody>
<tr>
<td>JACOBI CG</td>
<td>4N + 2·ITMAX</td>
</tr>
<tr>
<td>JACOBI SI</td>
<td>3N</td>
</tr>
<tr>
<td>SOR</td>
<td>N</td>
</tr>
<tr>
<td>SYMMETRIC SOR CG</td>
<td>6N + 2·ITMAX</td>
</tr>
<tr>
<td>SYMMETRIC SOR SI</td>
<td>4N</td>
</tr>
<tr>
<td>REDUCED SYSTEM CG</td>
<td>5N + 2·ITMAX</td>
</tr>
<tr>
<td>REDUCED SYSTEM SI</td>
<td>3N</td>
</tr>
</tbody>
</table>

For data transformation, AMATRX is required to be "number of equations" x 10 and BVECTOR is "number of equations" for reordered right-hand side.
MODULE NAME: YALE SPARSE

AUTHOR/DATE: Andrew H. Sherman, 9/15/78

INITIAL/FINAL INTERFACES: EQUATION SOLUTION/OUTPUT

MODULE FUNCTION: Solves Sparse linear systems

RESTRICTIONS ON USE: None

METHOD DESCRIPTION: An LU \((LDL^T)\) Factorization of the (symmetric) Sparse coefficient matrix is computed, and the resulting triangular systems are solved to obtain the solution. For further details see S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman, Yale Sparse Matrix Package. I: The Symmetric Codes, Report 112. II: The Non-symmetric Codes, Report 114, Computer Science, Yale University, 1977.

PARAMETERS: 
- \(NSP\) = amount of working storage
- \(DRV\) = \(\begin{cases} 
S & \text{Symmetric} \\
N & \text{Fast nonsymmetric} \\
G & \text{Compressed nonsymmetric} \\
T & \text{Storage conserving nonsymmetric}
\end{cases}\)

KEY-WORDS THAT AFFECT MODULE: SYMMET FORCES DRV DEFAULT TO S

STORAGE AND TIMING ESTIMATES: 
- Storage: \(NSP\) is amount of workspace
- Timing: \(O(N^{3/2})\) for \(N \times N\) 5-POINT linear system

COMPATIBLE WITH ELLPACK VERSIONS: Sept. 1978

NOTE: \(NSP\) is computed by default if not specified in this module or YALE MIN DEG module. (This may be a huge over estimate for small systems or too little for large ones in the April, 1980 version of ELLPACK.) If \(NSP\) is specified for more than one YALE module, the maximum value is used.
Module Name: YALE ENVELOPE

Author/Date: Andrew H. Sherman, 05/23/79

Initial/Final Interfaces: Equation Solution/Output

Module Function: Solves sparse (envelope) linear systems

Restrictions on Use: Matrix must be factorizable without partial pivoting

Method Description: The matrix is converted to envelope form and an $LU$ (LDL) factorization of the resulting (symmetric) matrix is computed. The triangular systems are then solved to obtain the solution. For further details see S.C. Eisenstat, M.C. Gursky, M.H. Schultz and A.H. Sherman, Yale Sparse Matrix Package, I: The symmetric codes, Report 112. II: The Non-symmetric codes, Report 114, Computer Science, Yale University, 1977.

Parameters: NSF = Amount of workspace storage

Keywords that affect module: SYMMET forces symmetric factorization. Otherwise the nonsymmetric factorization is used.

Storage and Timing Estimates:

Symmetric factorization: Workspace required = NSF = $2N + P$

Nonsymmetric factorization: Workspace required = NSF = $3N + P + Q$

Where $P = $ Size of envelope of strict upper triangle

$Q = $ Size of envelope of strict lower triangle

(If $MU$ is the upper half-Bandwidth, Then $P < N^*MU$)

(If $ML$ is the Lower half-Bandwidth, Then $Q < N^*ML$)

Note: NSF is computed by default if not specified in this module or in Yale Min Dog or in Yale sparse. This may be a huge overestimate for small systems and too little for large ones in the April '80 version of ELLPACK. If NSF is specified for more than one of these modules, the maximum value is used.