

July 1975

# **Efficient FORTRAN Subprograms for the Solution of Elliptic Partial Differential Equations**

Paul Swarztrauber  
Roland Sweet

---

ATMOSPHERIC TECHNOLOGY DIVISION

---

NATIONAL CENTER FOR ATMOSPHERIC RESEARCH  
BOULDER, COLORADO



**PREFACE**

This Technical Note is intended to provide scientists with a package of computer programs which make use of current methods for solving elliptic partial differential equations. While some discussion of these methods is included, program descriptions are presented separately for those who may wish only to use the package as a computational aid.

The work reported here is the product of a project that has been supported by NCAR over the past two years.



## FOREWORD

Computer models of atmospheric processes often require the numerical solution of elliptic partial differential equations. This is particularly true for models which make use of stream functions, velocity potentials, or vorticity equations, or in which the pressure of an incompressible fluid is computed. The numerical solution of elliptic equations can be a formidable programming task. Also, the equations are often time-dependent, requiring repeated solutions and, hence, considerable computing resources. Efficient, reliable, and well-documented computer subroutines for solving such equations are needed. With recent advances in computing methods [1, 2, 4, 5, 7, 9, 10], it became apparent to the authors that a very large class of elliptic equations could be solved rapidly and with minimal storage. Of particular importance was the fact that, as a result of work on singular problems [6, 8], this class was free of special cases for which solutions did not exist.

This report contains seven chapters, each describing one of the seven FORTRAN subroutines, and an appendix dealing with solutions of linear systems of equations. The first five subroutines solve a Helmholtz equation in various coordinate systems: they are, respectively, Chapter I, subroutine PWSCRT, Cartesian coordinates  $x$  and  $y$ ; Chapter II, subroutine PWSPLR, polar coordinates  $r$  and  $\theta$ ; Chapter III, subroutine PWSCYL, cylindrical coordinates  $r$  and  $z$ ; Chapter IV, subroutine PWSCSP, interior spherical coordinates,  $r$  and  $\theta$  (colatitude); and Chapter V, subroutine PWSSSP, surface spherical coordinates  $\theta$  and  $\phi$  (longitude).

Although these programs solve only two-dimensional problems, they can easily be adapted for use in three-dimensional problems by Fourier transforming in the third variable and using the parameter in the Helmholtz equation. This application is discussed further in the sections entitled Three-Dimensional Problems in each of the first five chapters.

In addition to the description of the FORTRAN subroutines, the first five chapters contain the finite difference approximations which are used for the Helmholtz equation and the boundary conditions. Any special equations at singular points of the Helmholtz equation (e.g., at the origin  $r = 0$ ) are also presented. Approximations of second-order accuracy are used throughout. Each of these five chapters contains a section entitled Singular Problems which augments the Appendix in describing how solutions can be obtained in the least-squares sense when a solution does not exist in the usual sense.

Each of these five subroutines calls one of the last two subroutines (POIS and BLKTRI), which are described in the remaining chapters. These subroutines can be used to solve a more general class of equation. Chapter VI describes subroutine POIS, which can be used to solve finite difference approximations to an equation of the form:

$$a(x) \frac{\partial^2 u}{\partial x^2} + b(x) \frac{\partial u}{\partial x} + c(x)u + \frac{\partial^2 u}{\partial y^2} = g(x, y).$$

Chapter VII describes subroutine BLKTRI, which can be used to solve finite difference approximations to the general separable equation:

$$a(x) \frac{\partial^2 u}{\partial x^2} + b(x) \frac{\partial u}{\partial x} + c(x)u + d(y) \frac{\partial^2 u}{\partial y^2} + e(y) \frac{\partial u}{\partial y} + f(y)u = g(x, y).$$

Although BLKTRI can solve problems that POIS solves, POIS should be used where possible since it is faster.

The last two chapters contain an extensive description of how these two subroutines can be used to solve elliptic partial differential equations that cannot be solved by the other programs. This description provides a guide to the user who must write a driver program that discretizes the equation and computes the quantities which are needed for input to POIS or BLKTRI.

The complete set of subroutines is available on punched cards from:

Information Services Office, Computing Facility  
National Center for Atmospheric Research  
Box 3000  
Boulder, Colorado 80303

To assist the user in installing the set at his computing center we provide copies of the sample problems given in each chapter. We have made every attempt to check out each program by testing all combinations of input parameters, but we cannot guarantee absolutely that the programs are error-free. Should an error be detected, please contact one of the authors at NCAR for assistance.



## CONTENTS

Preface . . . . .	iii
Foreword . . . . .	v
I. Subroutine PWSCRT . . . . .	1
II. Subroutine PWSPLR . . . . .	17
III. Subroutine PWSCYL . . . . .	37
IV. Subroutine PWSCSP . . . . .	60
V. Subroutine PWSSSP . . . . .	81
VI. Subroutine POIS . . . . .	96
VII. Subroutine BLKTRI . . . . .	115
Appendix . . . . .	133
References . . . . .	139



## I. SUBROUTINE PWSCRT

In this chapter we describe the subroutine PWSCRT which solves the standard five-point finite difference approximation to the Helmholtz equation in cartesian coordinates:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \lambda u = f(x,y) .$$

### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL PWSCRT (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,  
             BDC,BDD,ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
```

where the arguments are defined as:

- |                 |   |
|-----------------|---|
| <u>On Input</u> | INTL  |
|                 | = 0 on initial entry to PWSCRT or if N or NBDCND are<br>changed from previous call. |
|                 | - = 1 if N and NBDCND are unchanged from previous call<br>to PWSCRT.                |

Note: A call with INTL = 1 is about 1% faster than  
a call with INTL = 0.

A,B

The range of x; i.e.,  $A \leq x \leq B$ . A must be less than B.

M

The number of panels into which the interval (A,B) is subdivided. Hence, there will be M + 1 grid points in the x-direction given by  $x_I = A + (I-1)\Delta x$  for I = 1,2,...,M + 1, where  $\Delta x = (B-A)/M$  is the panel width.

MBDCND

Indicates the type of boundary conditions at  $x = A$  and  $x = B$ .

- = 0 if the solution is periodic in  $x$ ; i.e.,  $U(1,J) = U(M + 1,J)$ .
- = 1 if the solution is specified at  $x = A$  and  $x = B$ .
- = 2 if the solution is specified at  $x = A$  and the derivative of the solution with respect to  $x$  is specified at  $x = B$ .
- = 3 if the derivative of the solution with respect to  $x$  is specified at  $x = A$  and  $x = B$ .
- = 4 if the derivative of the solution with respect to  $x$  is specified at  $x = A$  and the solution is specified at  $x = B$ .

BDA

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $x$  at  $x = A$ . When MBDCND = 3 or 4,

$$BDA(J) = \frac{\partial u}{\partial x} (A, y_J) , J = 1, 2, \dots, N + 1 .$$

When MBDCND has any other value, BDA is a dummy variable.

BDB

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $x$  at  $x = B$ . When MBDCND = 2 or 3,

$$BDB(J) = \frac{\partial u}{\partial x} (B, y_J) , J = 1, 2, \dots, N + 1 .$$

When MBDCND has any other value BDB is a dummy variable.

C,D

The range of  $y$ ; i.e.,  $C \leq y \leq D$ . C must be less than D.

N

The number of panels into which the interval (C,D) is subdivided. Hence, there will be N + 1 grid points in the y-direction given by  $y_J = C + (J-1)\Delta y$  for  $J = 1, 2, \dots, N + 1$ , where  $\Delta y = (D-C)/N$  is the panel width. N must be of the form  $2^p 3^q 5^r$  where p, q, and r are any non-negative integers. N must be greater than 2.

NBDCND

Indicates the type of boundary conditions at  $y = C$  and  $y = D$ .

- = 0 if the solution is periodic in y; i.e.,  $U(I,1) = U(I, N + 1)$ .
- = 1 if the solution is specified at  $y = C$  and  $y = D$ .
- = 2 if the solution is specified at  $y = C$  and the derivative of the solution with respect to y is specified at  $y = D$ .
- = 3 if the derivative of the solution with respect to y is specified at  $y = C$  and  $y = D$ .
- = 4 if the derivative of the solution with respect to y is specified at  $y = C$  and the solution is specified at  $y = D$ .

BDC

A one-dimensional array of length M + 1 that specifies the values of the derivative of the solution with respect to y at  $y = C$ . When NBDCND = 3 or 4,

$$BDC(I) = \frac{\partial u}{\partial y} (x_I, C) , \quad I = 1, 2, \dots, M + 1 .$$

When NBDCND has any other value, BDC is a dummy variable.

BDD

A one-dimensional array of length M + 1 that specifies the values of the derivative of the solution with respect to y at  $y = D$ . When NBDCND = 2 or 3,

$$BDD(I) = \frac{\partial u}{\partial y} (x_I, D) , \quad I = 1, 2, \dots, M + 1 .$$

When NBDCND has any other value, BDD is a dummy argument.

#### ELMBDA

The constant  $\lambda$  in the Helmholtz equation. If  $\lambda > 0$ , a solution may not exist. However, PWSCRT will attempt to find a solution.

#### F

A two-dimensional array which specifies the values of the right side of the Helmholtz equation and boundary values (if any). For  $I = 2,3,\dots,M$  and  $J = 2,3,\dots,N$

$$F(I,J) = f(x_I, y_J) .$$

On the boundaries F is defined by

<u>MBDCND</u>	<u><math>F(1,J)</math></u>	<u><math>F(M+1,J)</math></u>
0	$f(A, y_J)$	$f(A, y_J)$
1	$u(A, y_J)$	$u(B, y_J)$
2	$u(A, y_J)$	$f(B, y_J) \quad J = 1, 2, \dots, N + 1$
3	$f(A, y_J)$	$f(B, y_J)$
4	$f(A, y_J)$	$u(B, y_J)$

  

<u>NBDCND</u>	<u><math>F(I,1)</math></u>	<u><math>F(I,N+1)</math></u>
0	$f(x_I, C)$	$f(x_I, C)$
1	$u(x_I, C)$	$u(x_I, D)$
2	$u(x_I, C)$	$f(x_I, D) \quad I = 1, 2, \dots, M + 1$
3	$f(x_I, C)$	$f(x_I, D)$
4	$f(x_I, C)$	$u(x_I, D)$

F must be dimensioned at least  $(M + 1) \times (N + 1)$ .

#### IDIMF

The row (or first) dimension of the array F as it appears in the program calling PWSCRT. This parameter is used to specify the variable dimension of F. IDIMF must be at least  $M + 1$ .

W

A one-dimensional array that must be provided by the user for work space. The length of W must be at least  $6(N + 1) + 8(M + 1)$ .

On Output

F

Contains the solution  $U(I,J)$  of the finite difference approximation for the grid point  $(x_I, y_J)$ ,  
 $I = 1, 2, \dots, M + 1$ ,  $J = 1, 2, \dots, N + 1$ .

PERTRB

If a combination of periodic or derivative boundary conditions is specified for a Poisson equation ( $\lambda = 0$ ), a solution may not exist. PERTRB is a constant, calculated and subtracted from F, which ensures that a solution exists. PWSCRT then computes this solution, which is a least squares solution to the original approximation. This solution is not unique.

IERROR

An error flag that indicates invalid input parameters. Except for numbers 0 and 6, a solution is not attempted.

- = 0 no error
- = 1  $A \leq B$
- = 2  $MBDCND < 0$  or  $MBDCND > 4$
- = 3  $C \geq D$
- = 4  $N \neq 2^P 3^Q 5^R$  or  $N \leq 2$
- = 5  $NBDCND < 0$  or  $NBDCND > 4$
- = 6  $\lambda > 0$
- = 7  $IDIMF < M + 1$

Since this is the only means of indicating a possibly incorrect call to PWSCRT, the user should test IERROR after the call.

W

Contains intermediate values that must not be destroyed if PWSCRT will be called again with INTL = 1.

Entry Points

PWSCRT, NCHECK

Special Conditions      None

Common Blocks      None

I/O      None

Precision      Single

Originator      Roland Sweet

Space Required      693<sub>10</sub> locations on the NCAR Control Data 7600.

Timing and Accuracy      The execution time is proportional to MN log<sub>2</sub>N and is tabulated below for the NCAR Control Data 7600 computer. To test the accuracy of the method a uniform random number generator was used to create an array V(I,J), where  $0 \leq V(I,J) \leq 1$ . An array F(I,J) was then computed by differencing V(I,J) in double precision, using the difference equations that correspond to the sample problem at the end of this chapter. With F(I,J) as a right side, subroutine PWSCRT was used to compute a solution U(I,J). The maximum absolute difference between U(I,J) and V(I,J) is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

M	N	Execution Time (msec)	Max. Absolute Difference
32	32	49	$5.0 \times 10^{-13}$
64	64	211	$1.0 \times 10^{-12}$
128	128	926	$6.1 \times 10^{-12}$

The execution times should be doubled if MBDCND = 0. They should be increased 30% if NBDCND = 3 and decreased 30% for any other value of NBDCND.

Portability American National Standards Institute FORTRAN with no machine-dependent constants.

Required Resident Routines None

#### DIFFERENCE APPROXIMATIONS

Subroutine PWSCRT sets up the standard five-point finite difference approximation to the Helmholtz equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \lambda u = f(x, y) , \quad (1)$$

on the rectangle  $A < x < B$ ,  $C < y < D$  and incorporates the given boundary data to form a linear system of equations, which is solved by subroutine POIS (see Chapter VI).

To describe the system of equations formed, we define a grid of points  $(x_i, y_j)$  by selecting integers M and N and defining

$$x_i = A + (i-1)\Delta x , \quad i = 1, 2, \dots, M+1 , \\ y_j = C + (j-1)\Delta y , \quad j = 1, 2, \dots, N+1 , \quad (2)$$

where  $\Delta x = \frac{B-A}{M}$  and  $\Delta y = \frac{D-C}{N}$ . If we now approximate each derivative of Eq. (1) by the usual second-order central difference and denote the approximation to  $u(x_i, y_j)$  by  $u_{i,j}$ , we get

$$\frac{1}{\Delta x^2} \left[ u_{i-1,j} - 2u_{i,j} + u_{i+1,j} \right] + \frac{1}{\Delta y^2} \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \lambda u_{i,j} = f(x_i, y_j) . \quad (3)$$

Eq. (3) is the basic equation used to determine the unknowns  $u_{i,j}$ .

Near the boundaries this equation is modified by the boundary conditions. The three types of boundary conditions which may be assumed are (say, at  $x = A$ ):

1. Solution specified:

We are given a function  $g$  such that

$$u(A, y) = g(y), \quad C \leq y \leq D .$$

In this case,  $u_{1,j}$  is known

$$u_{1,j} = g(y_j) , \quad j = 1, 2, \dots, N + 1 , \quad (4)$$

and the  $i$ -index of the unknowns  $u_{i,j}$  begins with  $i = 2$ . Incorporating Eq. (4) into Eq. (3) for  $i = 2$ , we get

$$\begin{aligned} & \frac{1}{\Delta x^2} \left[ -2u_{2,j} + u_{3,j} \right] + \frac{1}{\Delta y^2} \left[ u_{2,j-1} - 2u_{2,j} + u_{2,j+1} \right] + \lambda u_{2,j} \\ & = f(x_2, y_j) - \frac{1}{\Delta x^2} g(y_j) , \end{aligned}$$

which illustrates how the boundary data enter into the right side of the equation.

2. Derivative of the solution with respect to  $x$  specified:

We are given a function  $h(y)$  such that

$$\frac{\partial u}{\partial x}(A, y) = h(y) . \quad (5)$$

Since the solution is unknown at  $x = A$ , the  $i$ -index of the unknowns  $u_{i,j}$  begins with  $i = 1$ . We then assume that Eq. (1) holds at  $x = A$  and that the unknown  $u_{1,j}$  is defined by Eq. (3) with  $i = 1$ :

$$\begin{aligned} \frac{1}{\Delta x^2} \left[ u_{0,j} - 2u_{1,j} + u_{2,j} \right] + \frac{1}{\Delta y^2} \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \lambda u_{1,j} \\ = f(A, y_j) . \end{aligned} \quad (6)$$

This assumption requires the introduction of a virtual point  $(x_0, y_j)$ , which lies just outside the boundary. The unknown  $u_{0,j}$  is eliminated from Eq. (6) by approximating Eq. (5) with a second-order central difference formula to get

$$u_{0,j} = u_{2,j} - 2\Delta x h(y_j) . \quad (7)$$

Combining Eqs. (6) and (7), we arrive at the defining equation for  $u_{1,j}$

$$\begin{aligned} \frac{1}{\Delta x^2} \left[ -2u_{1,j} + 2u_{2,j} \right] + \frac{1}{\Delta y^2} \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \lambda u_{1,j} \\ = f(A, y_j) + \frac{2}{\Delta x} h(y_j) . \end{aligned}$$

### 3. Solution periodic in x:

In this case we assume that

$$u(A + x, y) = u(B + x, y)$$

which leads us to the conditions

$$u_{0,j} = u_{M,j} \quad \text{and} \quad u_{M+1,j} = u_{1,j} . \quad (8)$$

Again, we assume that Eq. (1) holds at  $x = A$ . We see that  $u_{0,j}, u_{1,j}, \dots, u_{M+1,j}$  are all unknown, but due to Eq. (8), a complete, nonredundant set of unknowns is  $u_{1,j}, u_{2,j}, \dots, u_{M,j}$ . The defining equation for  $u_{1,j}$ , Eq. (6), is modified using Eq. (8)

to obtain

$$\frac{1}{\Delta x^2} \left[ u_{M,j} - 2u_{1,j} + u_{2,j} \right] + \frac{1}{\Delta y^2} \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \lambda u_{1,j} = f(A, y_j) .$$

Similarly, the equation for  $u_{M,j}$  becomes

$$\frac{1}{\Delta x^2} \left[ u_{M-1,j} - 2u_{M,j} + u_{1,j} \right] + \frac{1}{\Delta y^2} \left[ u_{M,j-1} - 2u_{M,j} + u_{M,j+1} \right] + \lambda u_{M,j} = f(x_M, y_j) .$$

Notice that in this case the structure of the equations has changed considerably.

When periodicity in  $x$  is assumed, it is not necessary that  $B-A = 2\pi$ . However, it is likely that  $k(B-A) = 2\pi$  for some integer  $k$  which may correspond to the solution for the  $k$ th wave number. This permits solutions for high wave numbers with less computation.

By examining the input parameters, subroutine PWSCRT determines which  $u_{i,j}$  are unknowns, sets up the equations defining them (as illustrated above), and incorporates the given boundary data. The final linear system of equations is of the form which subroutine POIS can solve.

#### SINGULAR PROBLEMS

When the user specifies MBDCND and NBDCND = 0 or 3, the resulting linear system is singular. To ensure that a solution to the problem exists, the right side of the equation is perturbed by subtracting the constant PERTRB. (The development of this method is covered in the appendix.)

The vector  $h$  in the analysis given in the Appendix has a component

$$h_{i,j} = h_{x_i} \cdot h_{y_j}$$

corresponding to each unknown  $u_{i,j}$ . From the above discussion we know the range of the indices  $i$  and  $j$  will be  $i = 1, 2, \dots, IF$  and  $j = 1, 2, \dots, JF$ . The following tables define the quantities IF,  $h_{x_i}$ , JF, and  $h_{y_j}$ .

MBDCND	IF	$h_{x_i}$ ( $1 < i < IF$ )	$h_{x_1} = h_{x_{IF}}$
0	M	1	1
3	M+1	2	1

NBDCND	JF	$h_{y_j}$ ( $1 < j < JF$ )	$h_{y_1} = h_{y_{JF}}$
0	N	1	1
3	N+1	2	1

The restriction  $h^T f = 0$  (given in the Appendix) takes the form

$$h^T f = \sum_{i=1}^{IF} \sum_{j=1}^{JF} h_{x_i} h_{y_j} f_{i,j} = 0 \quad , \quad (9)$$

where  $f_{i,j}$  is the right side of Eq. (3) adjusted by the boundary data (if any). The constant PERTRB, defined as

$$\text{PERTRB} = \frac{h^T f}{h^T e}$$

is subtracted from the right side of the equation to ensure that the system is consistent. In this case

$$h^T e = \left[ 2 + h_{x_2}(IF-2) \right] \left[ 2 + h_{y_2}(JF-2) \right] \quad (10)$$

so PERTRB is just the quotient of Eqs. (9) and (10).

The solution that PWSCRT returns to the user is not unique since that solution plus any constant is also a solution. The solution can be checked by differencing it and comparing with the right-side f. Note, however, that the results will differ by the constant PERTRB.

### THREE-DIMENSIONAL PROBLEMS

This subroutine may be used in conjunction with a fast Fourier transform (FFT) routine to solve three-dimensional Poisson equations. Suppose one wants to solve the equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(x, y, z) \quad (11)$$

on the parallelepiped  $A < x < B$ ,  $C < y < D$ ,  $E < z < F$ , with the boundary conditions

$$u(A, y, z) = g(y, z), \quad u(B, y, z) = h(y, z), \quad (12a)$$

$$u(x, C, z) = u(x, D, z) = 0, \quad (12b)$$

$$u(x, y, E) = u(x, y, F) = 0. \quad (12c)$$

(Other boundary conditions may be treated similarly.)

Choosing M, N, and L to be positive integers, we define a grid

$$x_i = A + (i-1)\Delta x, \quad i = 1, 2, \dots, M+1,$$

$$y_j = C + (j-1)\Delta y, \quad j = 1, 2, \dots, N+1,$$

$$z_k = E + (k-1)\Delta z, \quad k = 1, 2, \dots, L+1,$$

where  $\Delta x = (B-A)/M$ ,  $\Delta y = (D-C)/N$ , and  $\Delta z = (F-E)/L$ , and replace Eq. (11) by the finite difference approximation

$$\begin{aligned} & \frac{1}{\Delta x^2} \left[ u_{i-1,j,k} - 2u_{i,j,k} + u_{i+1,j,k} \right] + \frac{1}{\Delta y^2} \left[ u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k} \right] \\ & + \frac{1}{\Delta z^2} \left[ u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1} \right] = f_{i,j,k}. \end{aligned} \quad (13)$$

If we now assume that

$$u_{i,j,k} = u_{i,j} \sin \omega k \Delta z$$

$$f_{i,j,k} = f_{i,j} \sin \omega k \Delta z$$

$$g_{j,k} = g_j \sin \omega k \Delta z$$

$$h_{j,k} = h_j \sin \omega k \Delta z ,$$

we find, after substituting these forms into Eq. (14) and dividing by  $\sin \omega k \Delta z$ , that the Fourier coefficients satisfy

$$\frac{1}{\Delta x^2} \left[ u_{i-1,j} - 2u_{i,j} + u_{i+1,j} \right] + \frac{1}{\Delta y^2} \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \lambda u_{i,j} = f_{i,j} , \quad (14)$$

where  $\lambda = \frac{2}{\Delta z^2} (\cos \omega \Delta z - 1)$ . On the boundaries we have from Eq. (13)

$$u_{i,j} = g_j , \quad u_{M+1,j} = h_j \quad (15)$$

$$u_{i,1} = 0 , \quad u_{i,N+1} = 0 .$$

Equation (14) with the boundary conditions of Eq. (15) may now be solved with subroutine PWSCRT.

In general,  $f_{i,j,k}$  will be a sum of sines and cosines. The Fourier coefficients for each term may be found by the method outlined above.

#### SAMPLE PROBLEM

In this section we illustrate the use of subroutine PWSCRT by presenting a FORTRAN program that solves the particular problem

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - 4u = [2 - (4+\pi^2/4)x^2] \cos \frac{\pi}{2}(y+1)$$

on the rectangle  $0 < x < 2$ ,  $-1 < y < 3$  with the boundary conditions

$$\left. \begin{array}{l} u(0,y) = 0 \\ \frac{\partial u}{\partial x}(2,y) = 4 \cos \frac{\pi}{2}(y+1) \end{array} \right\} \quad -1 \leq y \leq 3$$

and with  $u$  periodic in  $y$ .

The exact solution to this problem is  $u(x,y) = x^2 \cos \frac{\pi}{2}(y+1)$ .

The x-interval [0,2] will be divided into 100 panels and the y-interval [-1,3] into 80 panels. The finite difference approximation to this problem may be solved by the following program:

```
PROGRAM XAMPLE
C
C      PROGRAM TO ILLUSTRATE THE USE OF PWSCRT.
C
C      DIMENSION      F(120,100) ,BDB(81)           ,W(1294)           ,X(101),
C      1                 Y(81)
C
C      FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF. ALSO NOTE THAT W
C      IS DIMENSIONED 6*(N+1) + 8*(M+1).
C
C      IDIMF = 120
C      INTL = 0
C      A = 0.
C      B = 2.
C      M = 100
C      MBDCND = 2
C      C = -1.
C      D = 3.
C      N = 80
C      NBDCND = 0
C      ELMBDA = -4.

C
C      AUXILIARY QUANTITIES.
C
C      PI = 3.14159265358979
C      PIBY2 = PI/2.
C      PISQ = PI**2
C      MP1 = M+1
C      NP1 = N+1
C
C      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING
C      BOUNDARY DATA AND THE RIGHT SIDE OF THE HELMHOLTZ EQUATION.
C
C      DO 100 I=1,MP1
C          X(I) = (I-1)/50.
100 CONTINUE
      DO 105 J=1,NP1
          Y(J) = -1. + (J-1)/20.
105 CONTINUE
C
C      GENERATE BOUNDARY DATA.
C
```

```
DO 110 J=1,NP1
      BDB(J) = 4.*COS((Y(J)+1.)*PIBY2)
110 CONTINUE
C
C      BDA, BDC, AND BDD ARE DUMMY VARIABLES.
C
      DO 115 J=1,NP1
          F(1,J) = 0.
115 CONTINUE
C
C      GENERATE RIGHT SIDE OF EQUATION.
C
      DO 125 I=2,MP1
          DO 120 J=1,NP1
              F(I,J) = (2. - (4. + PISQ/4.)*X(I)**2)*COS((Y(J)+1.)*PIBY2)
120      CONTINUE
125 CONTINUE
      CALL PWSCRT (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,
1                  ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
C
C      COMPUTE DISCRETIZATION ERROR.  THE EXACT SOLUTION IS
C          U(X,Y) = X**2*COS((Y+1)*PIBY2)
C
      ERR = 0.
      DO 135 I=1,MP1
          DO 130 J=1,NP1
              Z = ABS(F(I,J)-X(I)**2*COS((Y(J)+1.)*PIBY2))
              IF (Z .GT. ERR) ERR = Z
130      CONTINUE
135 CONTINUE
      PRINT 1001 , IERROR,ERR
C
1001 FORMAT (///,9H IERROR =,I2,10X,22HDISCRETIZATION ERROR =,E12.5)
C
      END
```

After the CALL to PWSCRT, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed.

The output line is:

```
IERROR = 0           DISCRETIZATION ERROR = 5.35989E-04
```

We see that four digit accuracy has been achieved by this approximation. Note, however, the contrast between the discretization error and the observed roundoff error reported in the Timing and Accuracy subsection above.

## II. SUBROUTINE PWSPLR

In this chapter we describe the subroutine PWSPLR which solves a finite difference approximation to the Helmholtz equation in polar coordinates:

$$\frac{1}{r} \frac{\partial}{\partial r} (r \frac{\partial u}{\partial r}) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \lambda u = f(r, \theta).$$

### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL PWSPLR (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,  
             BDC,BDD,ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
```

where the arguments are defined as:

On Input            INTL

= 0 on initial entry to PWSPLR or if N or NBDCND  
are changed from previous call.

= 1 if N and NBDCND are unchanged from previous  
call to PWSPLR.

Note: A call with INTL = 1 is about 1% faster than  
a call with INTL = 0.

A,B

The range of r; i.e.,  $A \leq r \leq B$ . A must be less than  
B and A must be non-negative.

M

The number of panels into which the interval (A,B)  
is subdivided. Hence, there will be M + 1 grid  
points in the r-direction given by  $r_I = A + (I-1)\Delta r$ ,  
for  $I = 1, 2, \dots, M + 1$ , where  $\Delta r = (B - A)/M$  is the  
panel width.

MBDCND

Indicates the type of boundary condition at  $r = A$  and  $r = B$ .

- = 1 if the solution is specified at  $r = A$  and  $r = B$ .
- = 2 if the solution is specified at  $r = A$  and the derivative of the solution with respect to  $r$  is specified at  $r = B$ .
- = 3 if the derivative of the solution with respect to  $r$  is specified at  $r = A$  (see note below) and  $r = B$ .
- = 4 if the derivative of the solution with respect to  $r$  is specified at  $r = A$  (see note below) and the solution is specified at  $r = B$ .
- = 5 if the solution is unspecified at  $r = A = 0$  and the solution is specified at  $r = B$ .
- = 6 if the solution is unspecified at  $r = A = 0$  and the derivative of the solution with respect to  $r$  is specified at  $r = B$ .

Note: If  $A = 0$ , do not use  $MBDCND = 3$  or  $4$ , but instead use  $MBDCND = 1, 2, 5$ , or  $6$ .

BDA

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = A$ . When  $MBDCND = 3$  or  $4$ ,

$$BDA(J) = \frac{\partial u}{\partial r}(A, \theta_J), \quad J = 1, 2, \dots, N + 1.$$

When  $MBDCND$  has any other value,  $BDA$  is a dummy variable.

BDB

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = B$ . When  $MBDCND = 2, 3$ , or  $6$ ,

$$BDB(J) = \frac{\partial u}{\partial r}(B, \theta_J), \quad J = 1, 2, \dots, N + 1.$$

When MBDCND has any other value, BDB is a dummy variable.

C,D

The range of  $\theta$ ; i.e.,  $C \leq \theta \leq D$ . C must be less than D.

N

The number of panels into which the interval (C,D) is subdivided. Hence, there will be  $N + 1$  grid points in the  $\theta$ -direction given by  $\theta_J = C + (J-1)\Delta\theta$  for  $J = 1, 2, \dots, N + 1$ , where  $\Delta\theta = (D-C)/N$  is the panel width. N must be in the form  $2^p 3^q 5^r$  where p, q, and r are any non-negative integers. N must be greater than 2.

NBDCND

Indicates the type of boundary conditions at  $\theta = C$  and  $\theta = D$ .

- = 0 if the solution is periodic in  $\theta$ ; i.e.,  $U(I,1) = U(I,N + 1)$ .
- = 1 if the solution is specified at  $\theta = C$  and  $\theta = D$  (see note below).
- = 2 if the solution is specified at  $\theta = C$  and the derivative of the solution with respect to  $\theta$  is specified at  $\theta = D$  (see note below).
- = 3 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = C$  and  $\theta = D$ .
- = 4 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = C$  and the solution is specified at  $\theta = D$  (see note below).

Note: When NBDCND = 1, 2, or 4, do not use MBDCND = 5 or 6 (the former indicates that the solution is specified at  $r = 0$ ; the latter indicates the solution is unspecified at  $r = 0$ ). Use instead MBDCND = 1 or 2.

BDC

A one-dimensional array of length M + 1 that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = C$ . When NBDCND = 3 or 4,

$$BDC(I) = \frac{\partial u}{\partial \theta}(r_I, C), \quad I = 1, 2, \dots, M + 1.$$

When NBDCND has any other value, BDC is a dummy variable.

BDD

A one-dimensional array of length M + 1 that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = D$ . When NBDCND = 2 or 3,

$$BDD(I) = \frac{\partial u}{\partial \theta}(r_I, D), \quad I = 1, 2, \dots, M + 1.$$

When NBDCND has any other value, BDD is a dummy variable.

ELMBDA

The constant  $\lambda$  in the Helmholtz equation. If  $\lambda > 0$ , a solution may not exist. However, PWSPLR will attempt to find a solution.

F

A two-dimensional array that specifies the values of the right side of the Helmholtz equation and boundary values (if any). For  $I = 2, 3, \dots, M$  and  $J = 2, 3, \dots, N$

$$F(I, J) = f(r_I, \theta_J).$$

On the boundaries F is defined by

<u>MBDCND</u>	<u>F(1,J)</u>	<u>F(M+1,J)</u>
1	$u(A, \theta_J)$	$u(B, \theta_J)$
2	$u(A, \theta_J)$	$f(B, \theta_J)$
3	$f(A, \theta_J)$	$f(B, \theta_J)$
4	$f(A, \theta_J)$	$u(B, \theta_J)$ $J = 1, 2, \dots, N + 1$
5	$f(0, 0)$	$u(B, \theta_J)$
6	$f(0, 0)$	$f(B, \theta_J)$

<u>NBDCND</u>	<u>F(I,1)</u>	<u>F(I,N+1)</u>
0	$f(r_I, C)$	$f(r_I, C)$
1	$u(r_I, C)$	$u(r_I, D)$
2	$u(r_I, C)$	$f(r_I, D)$ $I = 1, 2, \dots, M + 1$
3	$f(r_I, C)$	$f(r_I, D)$
4	$f(r_I, C)$	$u(r_I, D)$

F must be dimensioned at least  $(M + 1) \times (N + 1)$ .

#### IDIMF

The row (or first) dimension of the array F as it appears in the program calling PWSPLR. This parameter is used to specify the variable dimension of F. IDIMF must be at least M + 1.

#### W

A one-dimensional array that must be provided by the user for work space. The length of W must be at least  $6(N + 1) + 8(M + 1)$ .

On Output

F

Contains the solution  $U(I,J)$  of the finite difference approximation for the grid point  $(r_I, \theta_J)$ ,  
 $I = 1, 2, \dots, M + 1$ ,  $J = 1, 2, \dots, N + 1$ .

PERTRB

If a combination of period, derivative, or unspecified boundary conditions is specified for a Poisson equation ( $\lambda = 0$ ), a solution may not exist. PERTRB is a constant, calculated and subtracted from F, which ensures that a solution exists. PWSPLR then computes this solution, which is a least squares solution to the original approximation. This solution is not unique.

IERROR

An error flag that indicates invalid input parameters. Except for numbers 0 and 11, a solution is not attempted.

- = 0 no error
- = 1  $A < 0$
- = 2  $A \geq B$
- = 3  $MBDCND < 1$  or  $MBDCND > 6$
- = 4  $C \geq D$
- = 5  $N \neq 2^P 3^Q 5^R$  or  $N \leq 2$
- = 6  $NBDCND < 0$  or  $> 4$
- = 7  $A = 0$ ,  $MBDCND = 3$  or  $4$
- = 8  $A > 0$ ,  $MBDCND \geq 5$
- = 9  $MBDCND \geq 5$ ,  $NBDCND \neq 0$  and  $NBDCND \neq 3$
- = 10  $IDIMF < M + 1$
- = 11  $\lambda > 0$

Since this is the only means of indicating a possibly incorrect call to PWSPLR, the user should test IERROR after the call.

W

Contains intermediate values that must not be destroyed if PWSPLR will be called again with INTL = 1.

Entry Points      PWSPLR, NCHECK

Special Conditions      None

Common Blocks      None

I/O      None

Precision      Single

Originator      Roland Sweet

Space Required       $990_{10}$  locations on the NCAR Control Data 7600.

Timing and Accuracy      The execution time is proportional to  $MN \log_2 N$  and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method, a uniform random number generator was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$ . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision, using the difference equations that correspond to the sample problem at the end of this chapter. With  $F(I,J)$  as a right side, subroutine PWSPLR was used to compute a solution  $U(I,J)$ . The maximum absolute difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

M	N	Execution Time (msec)	Max. Absolute Difference
32	32	65	$5.0 \times 10^{-12}$
64	64	272	$1.5 \times 10^{-11}$
128	128	1167	$1.4 \times 10^{-10}$

The execution times should be decreased by 30% if NBDCND = 0 and decreased by 50% for any other value of NBDCND.

Portability American National Standards Institute FORTRAN with no machine-dependent constants.

Required Resident Routines None

#### DIFFERENCE APPROXIMATIONS

Subroutine PWSPLR sets up a second-order finite difference approximation to the Helmholtz equation

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \lambda u = f(r, \theta), \quad (1)$$

on the portion of the disk  $A < r < B$ ,  $C < \theta < D$ , and incorporates the given boundary data to form a linear system of equations which is solved by subroutine POIS (see Chapter VI ).

To describe the linear system formed, we define a grid of points  $(r_i, \theta_j)$  by selecting integers M and N and defining

$$r_i = A + (i - 1)\Delta r, \quad i = 1, 2, \dots, M + 1 \quad (2)$$

$$\theta_j = C + (j - 1)\Delta \theta, \quad j = 1, 2, \dots, N + 1$$

where  $\Delta r = \frac{B-A}{M}$  and  $\Delta \theta = \frac{D-C}{N}$ . If we now approximate each derivative of Eq. (1) with a centered difference approximation and denote the

approximation to  $u(r_i, \theta_j)$  by  $u_{i,j}$ , we get (when  $r_i \neq 0$ )

$$\begin{aligned} & \frac{1}{\Delta r^2 r_i} \left[ (r_i + \frac{1}{2}\Delta r)(u_{i+1,j} - u_{i,j}) - (r_i - \frac{1}{2}\Delta r)(u_{i,j} - u_{i-1,j}) \right] \\ & + \frac{1}{\Delta \theta^2 r_i^2} \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \lambda u_{i,j} = f(r_i, \theta_j) \end{aligned}$$

or

$$\begin{aligned} & \frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,j} - \frac{2}{\Delta r^2} u_{i,j} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,j} \\ & + \left( \frac{1}{\Delta \theta r_i} \right)^2 \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \lambda u_{i,j} = f(r_i, \theta_j) \quad (3) \end{aligned}$$

Equation (3) is the basic equation used to determine the unknowns  $u_{i,j}$ .

Near the boundaries this equation is modified by the boundary conditions.

To describe the effect of the boundary conditions, let us first consider the case  $A > 0$  so that the origin is not included in the region of interest. At  $r = A$ , two different types of boundary conditions will be described.

1. Solution specified:

We are given a function  $g$  such that

$$u(A, \theta) = g(\theta), \quad C \leq \theta \leq D.$$

In this case,  $u_{1,j}$  is known

$$u_{1,j} = g(\theta_j), \quad j = 1, 2, \dots, N + 2, \quad (4)$$

and the  $i$ -index of the unknowns  $u_{i,j}$  begins with  $i = 2$ . Incorporating Eq. (4) into Eq. (3) for  $i = 2$ , we get

$$\begin{aligned} -\frac{2}{\Delta r^2} u_{2,j} + \frac{r_2 + \frac{1}{2}\Delta r}{\Delta r^2 r_2} u_{3,j} + \left(\frac{1}{\Delta \theta r_2}\right)^2 \left[ u_{2,j-1} - 2u_{2,j} + u_{2,j+1} \right] + \lambda u_{2,j} = \\ = f(r_2, \theta_j) - \frac{r_2 - \frac{1}{2}\Delta r}{\Delta r^2 r_2} g(\theta_j) \end{aligned}$$

which illustrates how the boundary data enter into the right side of the equation.

2. Derivative of the solution with respect to r specified:

We are given a function  $h(\theta)$  such that

$$\frac{\partial u}{\partial r} (A, \theta) = h(\theta). \quad (5)$$

Since the solution is unknown at  $r = A$ , the i-index of the unknowns  $u_{i,j}$  begins with  $i = 1$ . We then assume that Eq. (1) holds at  $r = A$  and that the unknown  $u_{1,j}$  is defined by Eq. (3) with  $i = 1$ :

$$\begin{aligned} \frac{r_1 - \Delta r}{\Delta r^2 r_1} u_{0,j} - \frac{2}{\Delta r^2} u_{1,j} + \frac{r_1 + \Delta r}{\Delta r^2 r_1} u_{2,j} \\ + \left( \frac{1}{r_1 \Delta \theta} \right)^2 \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \lambda u_{1,j} = f(A, \theta_j) \end{aligned} . \quad (6)$$

This assumption requires the introduction of a virtual point  $(r_0, \theta_j)$ , which lies just outside the boundary. The unknown  $u_{0,j}$  is eliminated from Eq. (6) by approximating Eq. (5) with a second-order central difference formula to get

$$u_{0,j} = u_{2,j} - 2\Delta r h(\theta_j). \quad (7)$$

Combining Eq. (6) and (7), we arrive at the defining equation for  $u_{1,j}$

$$\begin{aligned} - \frac{2}{\Delta r^2} u_{1,j} + \frac{2}{\Delta r^2} u_{2,j} + \left( \frac{1}{r_1 \Delta \theta} \right)^2 \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \lambda u_{1,j} \\ = f(A, \theta_j) + \frac{2r_1 - \Delta r}{\Delta r r_1} h(\theta_j). \end{aligned}$$

Note that the right side of the equation incorporates the boundary data and that the coefficient of  $u_{2,j}$  has changed.

These same boundary conditions may be specified at  $r = B$ .

At  $\theta = C$  and  $D$ , the two above boundary conditions may be specified, but a third type is also possible.

### 3. Solution periodic in $\theta$ :

In this case we assume that

$$u(r, C + \theta) = u(r, D + \theta)$$

which leads to the conditions

$$u_{i,0} = u_{i,N} \quad \text{and} \quad u_{i,N+1} = u_{i,1}. \quad (8)$$

Again, we assume that equation (1) holds at  $r = A$ . We see that  $u_{i,0}, u_{i,1}, \dots, u_{i,N+1}$  are unknowns, but due to Eq. (8) a complete nonredundant set of unknowns is  $u_{i,1}, u_{i,2}, \dots, u_{i,N}$ . The defining equation for  $u_{i,1}$ , Eq. (6), is modified using Eq. (8) to obtain

$$\begin{aligned} & \frac{r_i^{-1} \Delta r}{\Delta r^2 r_i} u_{i-1,1} - \frac{2}{\Delta r^2} u_{i,1} + \frac{r_i^{+1} \Delta r}{\Delta r^2 r_i} u_{i+1,1} \\ & + \left( \frac{1}{r_i \Delta \theta} \right)^2 \left[ u_{i,N} - 2u_{i,1} + u_{i,2} \right] + \lambda u_{i,1} = f(r_i, C) \end{aligned} .$$

Similarly, the equation for  $u_{i,N}$  becomes

$$\begin{aligned} & \frac{r_i^{-1} \Delta r}{\Delta r^2 r_i} u_{i-1,N} - \frac{2}{\Delta r^2} u_{i,N} + \frac{r_i^{+1} \Delta r}{\Delta r^2 r_i} u_{i+1,N} \\ & + \left( \frac{1}{r_i \Delta \theta} \right)^2 \left[ u_{i,N-1} - 2u_{i,N} + u_{i,1} \right] + \lambda u_{i,N} = f(r_i, \theta_N) \end{aligned} .$$

Notice that in this case the structure of the equation has changed considerably.

When periodicity in  $\theta$  is assumed, it is not necessary that  $D - C = 2\pi$ . However, it is likely that  $k(D - C) = 2\pi$  for some integer  $k$  which may correspond to the solution for the  $k$ th wave number. This permits solutions for high wave numbers with less computation.

When  $A = 0$  so that the origin is included in the region of interest, there is no change in the possible boundary conditions on the  $\theta$ -boundaries. However, one may not prescribe boundary condition 2 above at  $r = A$  but instead may prescribe condition 1 above or the following one.

#### 4. Solution unspecified at $r = A = 0$ :

This boundary condition may be specified when periodicity in  $\theta$  is specified so that the origin is really in the interior of the region of interest or when a derivative boundary condition is specified on both  $\theta$ -boundaries. Since the solution is unspecified and Eq. (3) cannot be used, we must develop an equation for  $u_{1,1}$ . We do this by multiplying Eq. (1) by  $r$  and integrating it over the sector of the disk  $0 \leq r \leq \frac{1}{2}\Delta r$ ,  $C \leq \theta \leq D$  to obtain

$$\begin{aligned} & \int_C^D \frac{\partial u}{\partial r}(\frac{1}{2}\Delta r, \theta) d\theta + \int_0^{\frac{1}{2}\Delta r} \frac{1}{r} \left[ \frac{\partial u}{\partial \theta}(r, D) - \frac{\partial u}{\partial \theta}(r, C) \right] dr \\ & + \int_0^{\frac{1}{2}\Delta r} \int_C^D r(\lambda u - f) d\theta dr = 0. \end{aligned} \quad (9)$$

Let us denote these integrals from left to right as  $I_1$ ,  $I_2$ , and  $I_3$ .

For  $I_1$  we first approximate the integrand by

$$\frac{\partial u}{\partial r}(\frac{1}{2}\Delta r, \theta) \approx \frac{1}{\Delta r} \left[ u(r_2, \theta) - u(0, 0) \right]$$

and then approximate the integral using the trapezoidal rule to get

$$I_1 \approx \frac{\Delta \theta}{2} \left[ \sum_{j=2}^N u_{2,j} + \frac{1}{2}u_{2,1} + \frac{1}{2}u_{2,N+1} - Nu_{1,1} \right]. \quad (10)$$

Integral  $I_2$  is zero when periodicity in  $\theta$  is specified. Now suppose the user specifies the derivative with respect to  $\theta$  on the two boundaries. Since

$$\frac{\partial u}{\partial \theta}(r, \theta) = r \left[ -\sin \theta \frac{\partial u}{\partial x} + \cos \theta \frac{\partial u}{\partial y} \right] ,$$

we see that  $\frac{\partial u}{\partial \theta}(0, \theta) = 0$ . Hence a linear approximation to  $\frac{\partial u}{\partial \theta}$  near  $r = 0$  can be found using a Taylor series, giving

$$\frac{\partial u}{\partial \theta}(r, \theta) \approx \frac{\partial u}{\partial \theta}(0, \theta) + r \left[ \frac{\frac{\partial u}{\partial \theta}(\Delta r, \theta) - \frac{\partial u}{\partial \theta}(0, \theta)}{\Delta r} \right] = \frac{r}{\Delta r} \frac{\partial u}{\partial \theta}(\Delta r, \theta).$$

Using this formula for  $I_2$  we get

$$\begin{aligned} I_2 &\approx \int_0^{\frac{1}{2}\Delta r} \frac{1}{\Delta r} \left[ \frac{\partial u}{\partial \theta}(\Delta r, D) - \frac{\partial u}{\partial \theta}(\Delta r, C) \right] dr \\ &\approx \frac{1}{2} \left[ \frac{\partial u}{\partial \theta}(r_2, \theta_{N+1}) - \frac{\partial u}{\partial \theta}(r_2, \theta_1) \right]. \end{aligned} \quad (11)$$

Note that the derivatives on the right side of Eq. (11) are input data.

Finally, a second order approximation to  $I_3$  may be obtained by replacing  $(\lambda u - f)(r, \theta)$  by  $(\lambda u - f)(0, 0)$  and integrating, or

$$\begin{aligned} I_3 &\approx \left( \frac{1}{8} \Delta r^2 \right) (D-C) \left[ \lambda u_{1,1} - f(0,0) \right] \\ &= \frac{N \Delta \theta \Delta r^2}{8} \left[ \lambda u_{1,1} - f(0,0) \right] . \end{aligned} \quad (12)$$

Combining Eqs. (10) through (12) and dividing by  $N\Delta\theta\Delta r^2/8$ , we obtain the formula for  $u_{1,1}$ , given by

$$\begin{aligned}
 & (\lambda - \frac{4}{\Delta r^2})u_{1,1} + \frac{4}{\Delta r^2 N} \left[ \sum_{j=2}^N u_{2,j} + \frac{1}{2}u_{2,1} + \frac{1}{2}u_{2,N+1} \right] \\
 & = f(0,0) - \frac{4}{N\Delta\theta\Delta r^2} \left[ BDD(2) - BDC(2) \right] \quad . \tag{13}
 \end{aligned}$$

By examining the input parameters, subroutine PWSPLR determines which  $u_{i,j}$  are unknowns, sets up the equations defining them (as illustrated above), and incorporates the given boundary data. The final linear system of equations is of the form that subroutine POIS can solve.

### SINGULAR PROBLEMS

When the user specifies a combination of periodic, derivative, or unspecified boundary conditions, the resulting linear system is singular. To ensure that a solution to the problem exists, the right side of the equation is perturbed by subtracting the constant PERTRB. (The development of this method is covered in the Appendix.)

The vector  $h$  in the analysis given in the Appendix has a component  $h_p$  corresponding to the unknown at the pole (if present) and a component

$$h_{ij} = h r_i h \theta_j , \quad i = IS, \dots, M+1, \quad j = 1, 2, \dots, JF$$

corresponding to each other unknown  $u_{ij}$ . The restriction  $h^T f = 0$  (given in the Appendix) takes the form

$$h^T f = h_p h \theta_2 f_p + \sum_{i=IS}^{M+1} \sum_{j=1}^{JF} h r_i h \theta_j \cdot f_{ij} = 0 \quad (14)$$

where  $f_p$  and  $f_{ij}$  are the values of the right side of the Helmholtz equation at the pole and at  $(r_i, \theta_j)$ , respectively, adjusted by the boundary data (if any). The values  $h_p$ ,  $h r_i$ ,  $h \theta_j$ ,  $IS$ , and  $JF$  depend on the specified boundary conditions and are defined by

MBDCND	$h_p$	IS	$h r_i$	$h r_i (2 \leq i \leq M)$	$h r_{M+1}$
3	0	1	$\frac{1}{2}r_i$	$r_i$	$\frac{1}{2}(r_M + \frac{1}{2}\Delta r)$
6	$\frac{N\Delta r}{8}$	2	0	$r_i$	$\frac{1}{2}(r_M + \frac{1}{2}\Delta r)$

NBDCND	JF	$h\theta_1$	$h\theta_j \ (2 \leq j < JF)$	$h_{JF}$
0	N	1	1	1
3	N+1	1	2	1

The constant PERTRB, calculated by

$$PERTRB = \frac{T_f}{h^T e} ,$$

is subtracted from the equation. In this case

$$h^T e = \left[ 2 + h\theta_2(JF-2) \right] \cdot \left[ h\theta_2 h_p + MA + \frac{1}{2} \frac{\Delta r}{IS} + \frac{1}{2} \Delta r(M-1)(M+1) \right] \quad (15)$$

so PERTRB is just the quotient of Eqs. (14) and (15).

The solution which PWSPLR returns to the user is not unique since that solution plus any constant is also a solution. If the user desires to check the solution by differencing and comparing it to the input data, he should remember that the results will differ by PERTRB.

### THREE-DIMENSIONAL PROBLEMS

This subroutine may be used in conjunction with a fast Fourier transform (FFT) routine to solve three dimensional Poisson equations. An example illustrating how one can do this is contained in the section, Three - Dimensional Problems, of Chapter 1. Since the development is identical, it will not be repeated here.

### SAMPLE PROBLEM

In this section we illustrate the use of subroutine PWSPLR by presenting a FORTRAN program which solves the particular problem:

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} = 16r^2 ,$$

on the quarter-disk  $0 < r < 1$ ,  $0 < \theta < \frac{1}{2}\pi$  with the boundary conditions:

$$u(1, \theta) = 1 - \cos 4\theta , \quad 0 \leq \theta \leq \frac{1}{2}\pi$$

and

$$\frac{\partial u}{\partial r}(r, 0) = \frac{\partial u}{\partial r}(r, \frac{1}{2}\pi) = 0 \quad , \quad 0 < r < 1.$$

Note that  $u$  at the origin is not specified and, hence, must be computed.  
The exact solution to this problem is:

$$u(r, \theta) = r^4(1 - \cos 4\theta) .$$

The  $r$ -interval  $[0,1]$  will be divided into 50 panels and the  $\theta$ -interval  $[0, \frac{1}{2}\pi]$  into 48 panels. The finite difference approximation to this problem may be solved by the following program:

```
PROGRAM XAMPLE
C
C      PROGRAM TO ILLUSTRATE THE USE OF PWSPLR.
C
C      DIMENSION      F(100,50)      ,BDC(51)      ,BDD(51)      ,W(702),
C      1            R(51)      ,THETA(49)
C
C      FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF. ALSO NOTE THAT W
C      IS DIMENSIONED 6*(N+1) + 8*(M+1).
C
C      IDIMF = 100
C      INTL = 0
C      A = 0.
C      B = 1.
C      M = 50
C      MBDCND = 5
C      C = 0.
C      PI = 3.14159265358979
C      D = PI/2.
C      N = 48
C      NBDCND = 3
C      ELMBDA = 0.
C
C      AUXILIARY QUANTITIES.
C
C      MP1 = M+1
C      NP1 = N+1
C
C      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING
C      BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.
C
C      DO 100 I=1,MP1
C          R(I) = (I-1)/50.
100 CONTINUE
```

```
DO 105 J=1,NP1
    THETA(J) = (J-1)*PI/96.
105 CONTINUE
C
C      GENERATE BOUNDARY DATA.
C
DO 110 I=1,MP1
    BDC(I) = 0.
    BDD(I) = 0.
110 CONTINUE
C
C      BDA AND BDB ARE DUMMY VARIABLES.
C
DO 115 J=1,NP1
    F(MP1,J) = 1.-COS(4.*THETA(J))
115 CONTINUE
C
C      GENERATE RIGHT SIDE OF EQUATION.
C
DO 125 I=1,M
    DO 120 J=1,NP1
        F(I,J) = 16.*R(I)**2
120    CONTINUE
125 CONTINUE
CALL PWSPLR (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,
1           ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
C
C      COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS
C      U(R,THETA) = R**4*(1 - COS(4*THETA))
C
ERR = 0.
DO 135 I=1,MP1
    DO 130 J=1,NP1
        Z = ABS(F(I,J)-R(I)**4*(1.-COS(4.*THETA(J))))
        IF (Z .GT. ERR) ERR = Z
130    CONTINUE
135 CONTINUE
PRINT 1001 , IERROR,ERR
STOP
C
1001 FORMAT (///,9H IERROR =,I2,10X,22HDISCRETIZATION ERROR =,E12.5)
C
END
```

After the CALL to PWSPLR, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed. The output line is:

IERROR = 0                    DISCRETIZATION ERROR = 6.19134E-04

We see from the output that the approximation has almost four digits of accuracy. Note, however, the contrast between the discretization error and the roundoff error reported in the Timing and Accuracy subsection above.

### III. SUBROUTINE PWSCYL

In this chapter we describe the subroutine PWSCYL which solves a finite difference approximation to the Helmholtz equation in cylindrical coordinates:

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{\partial^2 u}{\partial z^2} + \frac{\lambda}{r^2} u = f(r, z).$$

#### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL PWSCYL (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
```

where the arguments are defined as:

##### On Input

##### INTL

- = 0 on initial entry to PWSCYL or if N and NBDCND are changed from previous call
- = 1 if N and NBDCND are unchanged from previous call to PWSCYL

Note: A call with INTL = 1 is about 1% faster than a call with INTL = 0.

##### A,B

The range of  $r$ ; i.e.,  $A \leq r \leq B$ .  $A$  must be less than  $B$  and  $A$  must be non-negative.

##### M

The number of panels into which the interval  $(A, B)$  is subdivided. Hence, there will be  $M + 1$  grid points in the  $r$ -direction given by  $r_I = A + (I-1)\Delta r$ , for  $I = 1, 2, \dots, M + 1$ , where  $\Delta r = (B - A)/M$  is the panel width.

MBDCND

Indicates the type of boundary conditions at  $r = A$  and  $r = B$ .

- = 1 if the solution is specified at  $r = A$  and  $r = B$ .
- = 2 if the solution is specified at  $r = A$  and the derivative of the solution with respect to  $r$  is specified at  $r = B$ .
- = 3 if the derivative of the solution with respect to  $r$  is specified at  $r = A$  (see note below) and  $r = B$ .
- = 4 if the derivative of the solution with respect to  $r$  is specified at  $r = A$  (see note below) and the solution is specified at  $r = B$ .
- = 5 if the solution is unspecified at  $r = A = 0$  and the solution is specified at  $r = B$ .
- = 6 if the solution is unspecified at  $r = A = 0$  and the derivative of the solution with respect to  $r$  is specified at  $r = B$ .

Note: If  $A = 0$ , do not use MBDCND = 3 or 4, but instead use MBDCND = 1,2,5, or 6.

BDA

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = A$ . When MBDCND = 3 or 4,

$$BDA(J) = \frac{\partial u}{\partial r} (A, z_J) , \quad J = 1, 2, \dots, N + 1 .$$

When MBDCND has any other value, BDA is a dummy variable.

BDB

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = B$ . When MBDCND = 2,3, or 6,

$$BDB(J) = \frac{\partial u}{\partial r} (B, z_J) , \quad J = 1, 2, \dots, N + 1 .$$

When MBDCND has any other value BDB is a dummy variable.

C,D

The range of  $z$ ; i.e.,  $C \leq z \leq D$ .  $C$  must be less than  $D$ .

N

The number of panels into which the interval  $(C,D)$  is subdivided. Hence, there will be  $N + 1$  grid points in the  $z$ -direction given by  $z_J = C + (J-1)\Delta z$ , for  $J = 1, 2, \dots, N + 1$ , where  $\Delta z = (D - C)/N$  is the panel width.  $N$  must be of the form  $2^p 3^q 5^r$  where  $p, q$ , and  $r$  are any non-negative integers.  $N$  must be greater than 2.

NBDCND

Indicates the type of boundary conditions at  $z = C$  and  $z = D$ .

- = 0 if the solution is periodic in  $z$ ; i.e.,  $U(I,1) = U(I,N + 1)$ .
- = 1 if the solution is specified at  $z = C$  and  $z = D$ .
- = 2 if the solution is specified at  $z = C$  and the derivative of the solution with respect to  $z$  is specified at  $z = D$ .
- = 3 if the derivative of the solution with respect to  $z$  is specified at  $z = C$  and  $z = D$ .
- = 4 if the derivative of the solution with respect to  $z$  is specified at  $z = C$  and the solution is specified at  $z = D$ .

BDC

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution with respect to  $z$  at  $z = C$ . When NBDCND = 3 or 4,

$$BDC(I) = \frac{\partial u}{\partial z} (r_I, C) , \quad I = 1, 2, \dots, M + 1 .$$

When NBDCND has any other value, BDC is a dummy variable.

BDD

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution with respect to  $z$  at  $z = D$ . When  $NBDCND = 2$  or  $3$ ,

$$BDD(I) = \frac{\partial u}{\partial z}(r_I, D) \quad , \quad I = 1, 2, \dots, M + 1.$$

When  $NBDCND$  has any other value,  $BDD$  is a dummy variable.

ELMBDA

The constant  $\lambda$  in the Helmholtz equation. If  $\lambda > 0$ , a solution may not exist. However, PWSCYL will attempt to find a solution.  $\lambda$  must be zero when  $MBDCND = 5$  or  $6$  (for explanation, see Three-Dimensional Problems below).

F

A two-dimensional array that specifies the values of the right side of the Helmholtz equation and boundary data (if any). For  $I = 2, 3, \dots, M$  and  $J = 2, 3, \dots, N$

$$F(I, J) = f(r_I, z_J)$$

On the boundaries F is defined by

<u>MBDCND</u>	<u>F(1,J)</u>	<u>F(M+1,J)</u>
1	$u(A, z_J)$	$u(B, z_J)$
2	$u(A, z_J)$	$f(B, z_J)$
3	$f(A, z_J)$	$f(B, z_J)$
4	$f(A, z_J)$	$u(B, z_J)$
5	$f(0, z_J)$	$u(B, z_J)$
6	$f(0, z_J)$	$f(B, z_J)$

<u>NBDCND</u>	<u>F(I,1)</u>	<u>F(I,N+1)</u>
0	$f(r_I, C)$	$f(r_I, D)$
1	$u(r_I, C)$	$u(r_I, D)$
2	$u(r_I, C)$	$f(r_I, D)$
3	$f(r_I, C)$	$f(r_I, D)$
4	$f(r_I, C)$	$u(r_I, D)$

F must be dimensioned at least  $(M + 1) \times (N + 1)$ .

IDIMF

The row (or first) dimension of the array F as it appears in the program calling PWSCYL. This parameter is used to specify the variable dimension of F. IDIMF must be at least M + 1.

W

A one-dimensional array that must be provided by the user for work space. The length of W must be at least  $6(N + 1) + 8(M + 1)$ .

On Output

F

Contains the solution  $U(I,J)$  of the finite difference approximation for the grid point  $(r_I, z_J)$ ,  $I = 1, 2, \dots, M + 1$ ,  $J = 1, 2, \dots, N + 1$ .

PERTRB

If one specifies a combination of periodic, derivative, and unspecified boundary conditions for a Poisson equation ( $\lambda = 0$ ), a solution may not exist. PERTRB is a constant, calculated and subtracted from F, which ensures that a solution exists. PWSCYL then computes this solution, which is a least squares solution to the original approximation. This solution is not unique.

IERROR

An error flag which indicates invalid input parameters. Except for numbers 0 and 11, a solution is not attempted.

- = 0 no error
- = 1  $A < 0$
- = 2  $A \geq B$
- = 3  $MBDCND < 1$  or  $MBDCND > 6$
- = 4  $C \geq D$
- = 5  $N \neq 2^P 3^Q 5^R$  or  $N \leq 2$
- = 6  $NBDCND < 0$  or  $NBDCND > 4$
- = 7  $A = 0$ ,  $MBDCND = 3$  or  $4$
- = 8  $A > 0$ ,  $MBDCND \geq 5$

= 9  $A = 0, \lambda \neq 0, MBDCND \geq 5$   
= 10  $IDIMF < M + 1$   
= 11  $\lambda > 0$

Since this is the only means of indicating a possibly incorrect call to PWSCYL, the user should test IERROR after the call.

W

Contains intermediate values that must not be destroyed if PWSCYL will be called again with INTL = 1.

Entry Points PWSCYL, NCHECK

Special Conditions None

Common Blocks None

I/O None

Precision Single

Originator Roland Sweet

Space Required  $810_{10}$  locations on the NCAR Control Data 7600.

Timing and Accuracy The execution time is proportional to  $MN \log_2 N$  and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method, a uniform random number generator was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$ . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision, using the difference equations that correspond to the sample problem at the end of this chapter.

With  $F(I,J)$  as a right side, subroutine PWSCYL was used to compute a solution  $U(I,J)$ . The maximum absolute difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

M	N	Execution Time (msec)	Max. Absolute Difference
32	32	69	$3.8 \times 10^{-12}$
64	64	282	$7.9 \times 10^{-12}$
128	128	1192	$1.4 \times 10^{-11}$

The execution times should be decreased by 30% if NBDCND = 0 and decreased by 50% for any other value of NBDCND.

Portability

American National Standards Institute FORTRAN with no machine-dependent constants.

Required  
Resident  
Routines

None

DIFFERENCE APPROXIMATIONS

Subroutine PWSCYL sets up a second-order finite difference approximation to the Helmholtz equation

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{\partial^2 u}{\partial z^2} + \frac{\lambda}{r^2} u = f(r,z) \quad (1)$$

on the rectangle  $A \leq r \leq B$ ,  $C \leq z \leq D$ , and incorporates the given boundary data to form a linear system of equations which is solved by subroutine POIS (see Chapter VI). (The coefficient  $\frac{1}{r^2}$  of  $\lambda u$  is explained in the section Three-Dimensional Problems below).

To describe the linear system formed from the finite difference approximation, let us define a grid of points  $(r_i, z_j)$  by selecting integers  $M$  and  $N$  and defining

$$\begin{aligned} r_i &= A + (i - 1)\Delta r, & i = 1, 2, \dots, M + 1 \\ z_j &= C + (j - 1)\Delta z, & j = 1, 2, \dots, N + 1 \end{aligned} \quad (2)$$

where  $\Delta r = \frac{B - A}{M}$  and  $\Delta z = \frac{D - C}{N}$ . If we now approximate each derivative of Eq. (1) with a centered difference approximation and denote the approximation to  $u(r_i, z_j)$  by  $u_{i,j}$ , we get (when  $r_i \neq 0$ )

$$\begin{aligned} &\frac{1}{\Delta r^2 r_i} \left[ (r_i + \frac{1}{2}\Delta r)(u_{i+1,j} - u_{i,j}) - (r_i - \frac{1}{2}\Delta r)(u_{i,j} - u_{i-1,j}) \right] \\ &+ \frac{1}{\Delta z^2} \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \frac{\lambda}{r_i^2} u_{i,j} = f(r_i, z_j), \end{aligned}$$

or

$$\begin{aligned} &\frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,j} - \frac{2}{\Delta r^2} u_{i,j} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,j} \\ &+ \frac{1}{\Delta z^2} \left[ u_{i,j-1} - 2u_{i,j} + u_{i,j+1} \right] + \frac{\lambda}{r_i^2} u_{i,j} = f(r_i, z_j) \quad . \end{aligned} \quad (3)$$

Equation (3) is the basic equation used to determine the unknowns  $u_{i,j}$ . Near the boundaries this equation is modified by the boundary conditions.

To describe the effect of the boundary conditions let us consider first the case  $A > 0$ . At  $r = A$ , two different types of boundary conditions will be described.

1. Solution specified:

We are given a function  $g$  such that

$$u(A, z) = g(z), \quad C \leq z \leq D.$$

In this case,  $u_{1,j}$  is known

$$u_{1,j} = g(z_j), \quad j = 1, 2, \dots, N + 1, \quad (4)$$

and the  $i$ -index of the unknowns  $u_{i,j}$  begins with  $i = 2$ . Incorporating Eq. (4) into Eq. (3) for  $i = 2$ , we get

$$\begin{aligned} & -\frac{2}{\Delta r^2} u_{2,j} + \frac{r_2 + \frac{1}{2}\Delta r}{\Delta r^2 r_2} u_{3,j} + \frac{1}{\Delta z^2} \left[ u_{2,j-1} - 2u_{2,j} + u_{2,j+1} \right] + \frac{\lambda}{r_2^2} u_{2,j} \\ & = f(r_2, z_j) - \frac{r_2 - \frac{1}{2}\Delta r}{\Delta r^2 r_2} g(z_j), \end{aligned}$$

which illustrates how the boundary data enter into the right side of the equation.

2. Derivative of the solution with respect to  $r$  specified:

We are given a function  $h(z)$  such that

$$\frac{\partial u}{\partial r}(A, z) = h(z). \quad (5)$$

Since the solution is unknown at  $r = A$ , the  $i$ -index of the unknown  $u_{i,j}$  begins with  $i = 1$ . We assume that Eq. (1) holds at  $r = A$  and that the unknown  $u_{1,j}$  is defined by Eq. (3) with  $i = 1$ :

$$\frac{r_1 - \frac{1}{2}\Delta r}{\Delta r^2 r_1} u_{0,j} - \frac{2}{\Delta r^2} u_{1,j} + \frac{r_1 + \frac{1}{2}\Delta r}{\Delta r^2 r_1} u_{2,j} + \frac{1}{\Delta z^2} \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \frac{\lambda}{r_1^2} u_{1,j} = f(A, z_j) . \quad (6)$$

This assumption requires the introduction of a virtual point  $(r_0, z_j)$ , which lies just outside the boundary. The unknown  $u_{0,j}$  is eliminated from Eq. (6) by approximating Eq. (5) with a second-order central difference formula to get

$$u_{0,j} = u_{2,j} - 2\Delta r h(z_j) . \quad (7)$$

Combining Eqs. (6) and (7) we arrive at the defining equation for  $u_{1,j}$

$$-\frac{2}{\Delta r^2} u_{1,j} + \frac{2}{\Delta r^2} u_{2,j} + \frac{1}{\Delta z^2} \left[ u_{1,j-1} - 2u_{1,j} + u_{1,j+1} \right] + \frac{\lambda}{r_1^2} u_{1,j} = f(A, z_j) + \frac{2r_1 - \Delta r}{\Delta r r_1} h(z_j) .$$

Note that the right side of the equation reflects the boundary data and that the coefficient of  $u_{2,j}$  has changed.

These same boundary conditions may be specified at  $r = B$ .

At  $z = C$  and  $D$ , the two above boundary conditions may be specified, but a third type is also possible.

### 3. Solution periodic in z:

In this case we assume that

$$u(r, C + z) = u(r, D + z)$$

which leads to the conditions

$$u_{i,0} = u_{i,N} \quad \text{and} \quad u_{i,N+1} = u_{i,1} \quad . \quad (8)$$

Again, we assume that Eq. (1) holds at  $z = C$ . We see that  $u_{i,0}, u_{i,1}, \dots, u_{i,N+1}$  are unknowns, but due to Eq. (8) a complete nonredundant set of unknowns is  $u_{i,1}, u_{i,2}, \dots, u_{i,N}$ . The defining equation for  $u_{i,1}$ , Eq. (6), becomes

$$\begin{aligned} & \frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,1} - \frac{2}{\Delta r^2} u_{i,1} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,1} \\ & + \frac{1}{\Delta z^2} \left[ u_{i,N} - 2u_{i,1} + u_{i,2} \right] + \frac{\lambda}{r_i^2} u_{i,1} = f(r_i, C) \end{aligned} \quad .$$

Similarly, the equation for  $u_{i,N}$  becomes

$$\begin{aligned} & \frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,N} - \frac{2}{\Delta r^2} u_{i,N} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,N} \\ & + \frac{1}{\Delta z^2} \left[ u_{i,N-1} - 2u_{i,N} + u_{i,1} \right] + \frac{\lambda}{r_i^2} u_{i,N} = f(r_i, z_N) \end{aligned} \quad .$$

Notice that in this case the structure of the equations has changed considerably.

When periodicity in  $z$  is assumed, it is not necessary that  $D - C = 2\pi$ . However, it is likely that  $k(D - C) = 2\pi$  for some integer  $k$  which may correspond to the solution for the  $k$ th wave number. This permits solution for high wave numbers with less computation.

When  $A = 0$  the only boundary conditions that may be specified at  $r = A$  are condition 1 above or the following one.

4. Solution unspecified at  $r = A = 0$ :

This boundary condition may be specified when  $\lambda = 0$  (see section below, Three-Dimensional Problems). Since the solution is unspecified at  $r = 0$  and Eq. (3) cannot be used, we must develop a different equation for  $u_{1,j}$ .

The first term on the left side of Eq. (1) may be written as

$$\frac{1}{r} \frac{\partial}{\partial r} (r \frac{\partial u}{\partial r}) = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} . \quad (9)$$

If we assume that

$$\frac{\partial u}{\partial r} (0, z) = 0 , \quad (10)$$

the second term on the right side of Eq. (9) is indeterminate, in which case L'Hospital's rule can be used to obtain

$$\lim_{r \rightarrow 0} \frac{1}{r} \frac{\partial u}{\partial r} = \lim_{r \rightarrow 0} \frac{\partial^2 u}{\partial r^2} .$$

Hence, Eq. (1) at  $r = 0$  becomes

$$\frac{1}{r} \frac{\partial}{\partial r} (r \frac{\partial u}{\partial r}) + \frac{\partial^2 u}{\partial z^2} = 2 \frac{\partial^2 u}{\partial r^2} + \frac{\partial^2 u}{\partial z^2} = f(0, z) . \quad (11)$$

Introducing the virtual points  $(r_0, z_j)$  we write the second-order finite difference approximation to Eq. (10),

$$\frac{u_{2,j} - u_{0,j}}{2\Delta r} = 0 \quad (12)$$

and to Eq. (11)

$$2 \frac{u_{0,j} - 2u_{1,j} + u_{2,j}}{\Delta r^2} + \frac{u_{1,j-1} - 2u_{1,j} + u_{1,j+1}}{\Delta z^2} = f(0, z_j) \quad (13)$$

Combining Eqs. (12) and (13) to eliminate the unknown  $u_{0,j}$  gives

$$\frac{-4u_{1,j} + 4u_{2,j}}{\Delta r^2} + \frac{u_{1,j-1} - 2u_{1,j} + u_{1,j+1}}{\Delta z^2} = f(0, z_j) \quad (14)$$

which is the defining equation for the unknown  $u_{1,j}$ .

By examining the input parameters, subroutine PWSCYL determines which  $u_{i,j}$  are unknowns, sets up the equations defining them (as illustrated above), and incorporates the boundary data. The final linear system of equations is of the form that subroutine POIS can solve.

#### SINGULAR PROBLEMS

When the user specifies a combination of periodic, derivative, or unspecified boundary conditions, the resulting linear system of equations is singular. To ensure that a solution exists, the right side of the equation is perturbed by subtracting the constant PERTRB. (The development of this method is covered in the appendix.)

The vector  $h$  in the analysis given in the Appendix has a component

$$h_{ij} = h_r r_i h_z j$$

corresponding to each unknown  $u_{i,j}$ . The range of the indices  $i$  and  $j$  is  $i = 1, 2, \dots, M + 1$  and  $j = 1, 2, \dots, JF$ . The restriction  $h^T f = 0$  (given in the Appendix) takes the form

$$h^T f = \sum_{i=1}^{M+1} \sum_{j=1}^{JF} h_r i h_z j f_{i,j} = 0 , \quad (15)$$

where  $f_{i,j}$  is the value of the right side of the Helmholtz equation at  $(r_i, z_j)$  adjusted by the boundary data (if any). The values  $h_r i$ ,  $h_z j$ , and  $JF$  depend on the specified boundary conditions and are defined by

MBDCND	$h_r 1$	$h_r i (2 \leq i \leq M)$	$h_r M+1$
3	$\frac{1}{2}(r_2 - \frac{1}{2}\Delta r)$	$r_i$	$\frac{1}{2}(r_M + \frac{1}{2}\Delta r)$
6	$\frac{1}{4}(r_2 - \frac{1}{2}\Delta r)$	$r_i$	$\frac{1}{2}(r_M + \frac{1}{2}\Delta r)$

NBDCND	$JF$	$h_z 1$	$h_z j (2 \leq j < JF)$	$h_z JF$
0	$N$	1	1	1
3	$N + 1$	1	2	1

The constant PERTRB, calculated by

$$PERTRB = \frac{h^T f}{h^T e} ,$$

is subtracted from the equation. In this case

$$h^T e = \begin{cases} (\frac{1}{2}\Delta r \left[ 1 + (M-1)*(M+1) \right] + MA)(h_z 2 N), & \text{when MBDCND} = 3 \\ \frac{1}{2}\Delta r \left[ \frac{3}{4} + (M-1)*(M+1) \right] [h_z 2 N], & \text{when MBDCND} = 6 \end{cases} \quad (16)$$

so PERTRB is just the quotient of Eqs. (15) and (16).

The solution that PWSCYL returns to the user is not unique since that solution plus any constant is also a solution. If the user desires to check the solution by differencing and comparing it to the input data, he should note that the results should differ by PERTRB.

### THREE-DIMENSIONAL PROBLEMS

This subroutine may be used in conjunction with a fast Fourier transform (FFT) routine to solve three-dimensional Poisson equations. Suppose one wants to solve the three-dimensional Poisson equation in cylindrical coordinates

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2} + \frac{\partial^2 u}{\partial z^2} = f(r, \theta, z) , \quad (17)$$

in the cylinder  $0 < r < R$ ,  $0 < \theta < 2\pi$ ,  $C < z < D$  with the boundary conditions

$$u(R, \theta, z) = g(\theta, z) \quad (18)$$
$$u(r, \theta, C) = u(r, \theta, D) = 0$$

(Other boundary conditions may be treated similarly.)

We define a grid by choosing integers M, N, and L and setting

$$r_i = (i - 1)\Delta r , \quad i = 1, 2, \dots, M + 1$$

$$z_j = C + (j - 1)\Delta z , \quad j = 1, 2, \dots, N + 1$$

$$\theta_k = (k - 1)\Delta\theta , \quad k = 1, 2, \dots, L + 1$$

where  $\Delta r = \frac{R}{M}$ ,  $\Delta z = \frac{D-C}{N}$ , and  $\Delta\theta = \frac{2\pi}{L}$ , and replacing Eq. (17) by the finite difference approximation

$$\begin{aligned} & \frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,j,k} - \frac{2}{\Delta r^2} u_{i,j,k} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,j,k} \\ & + \frac{1}{\Delta z^2} \left[ u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k} \right] \\ & + \frac{1}{\Delta\theta^2 r_i^2} \left[ u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1} \right] = f(r_i, \theta_k, z_j) \end{aligned} \quad (19)$$

when  $i \neq 1$ . An equation for  $r = 0$  may be developed by multiplying Eq. (17) by  $r$  and integrating it over the disk  $0 \leq r \leq \frac{1}{2}r$ ,  $0 \leq \theta \leq 2\pi$  to get

$$\frac{\Delta r}{2} \int_0^{2\pi} \frac{\partial u}{\partial r} (\frac{1}{2}\Delta r, \theta, z) d\theta + \int_0^{\frac{1}{2}\Delta r} \int_0^{2\pi} r \left[ \frac{\partial^2 u}{\partial z^2} - f \right] d\theta dr = 0 \quad (20)$$

The integral on the left may be approximated by the trapezoidal rule and the integrand by a central difference, yielding

$$\frac{\Delta r}{2} \int_0^{2\pi} \frac{\partial u}{\partial r} (\frac{1}{2}\Delta r, \theta, z) d\theta \approx -\pi u(0, 0, z) + \frac{\pi}{N} \sum_{k=1}^N u(\Delta r, k\Delta\theta, z). \quad (21)$$

The second integral in Eq. (20) is approximated by

$$\begin{aligned} & \int_0^{\frac{1}{2}\Delta r} \int_0^{2\pi} r \left[ \frac{\partial^2 u}{\partial z^2} - f \right] d\theta dr \approx \left[ \frac{\partial^2 u}{\partial z^2} (0, 0, z) - f(0, 0, z) \right] \int_0^{\frac{1}{2}\Delta r} \int_0^{2\pi} r d\theta dr \\ & = \frac{\pi}{4} \Delta r^2 \left[ \frac{\partial^2 u}{\partial z^2} (0, 0, z) - f(0, 0, z) \right]. \end{aligned} \quad (22)$$

At the grid point  $(r_1, z_j, \theta_k) = (0, z_j, 0)$  we take as the equation the sum of Eqs. (21) and (22), or

$$\begin{aligned} & \frac{4}{\Delta r^2} \left[ -u_{1,j,k} + \frac{1}{N} \sum_{k=1}^N u_{2,k,j} \right] + \frac{1}{\Delta z^2} \left[ u_{1,j-1,k} - 2u_{1,j,k} + u_{1,j+1,k} \right] \\ & = f(0, 0, z_j) \end{aligned} \quad , \quad (23)$$

where we have made the usual approximation to  $\frac{\partial^2 u}{\partial z^2}$ . The boundary conditions, Eq. (18), become

$$\begin{aligned} u_{M+1,j,k} &= g(\theta_k, z_j) \quad j = 1, 2, \dots, N+1; \quad k = 1, 2, \dots, L+1 \\ u_{i,1,k} &= u_{i,N+1,k} = 0 \quad , \quad i = 1, 2, \dots, M+1, \quad k = 1, 2, \dots, L+1 \end{aligned} \quad (24)$$

To solve the linear system of Eqs. (10), (23), and (24), we write the solution and data as a Fourier series in  $\theta$ ; i.e.,

$$u_{i,j,k} = \sum_{m=0}^{\frac{1}{2}L} u_{i,j,m}^{(1)} \cos m\theta_k + u_{i,j,m}^{(2)} \sin m\theta_k \quad (25a)$$

$$f(r_i, \theta_k, z_j) = \sum_{m=0}^{\frac{1}{2}L} f_{i,j,m}^{(1)} \cos m\theta_k + f_{i,j,m}^{(2)} \sin m\theta_k \quad (25b)$$

$$g(\theta_k, z_j) = \sum_{m=0}^{\frac{1}{2}L} g_{j,m}^{(1)} \cos m\theta_k + g_{j,m}^{(2)} \sin m\theta_k . \quad (25c)$$

Given the values of  $f$  and  $g$ , the Fourier coefficients in Eqs. (25b) and (25c) are computed using an FFT routine. Substituting Eqs. (25) into Eqs. (19) and (24) and equating coefficients of sines and cosines, we get

$$\begin{aligned} \frac{r_i - \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i-1,j,m}^{(\ell)} - \frac{2}{\Delta r^2} u_{i,j,m}^{(\ell)} + \frac{r_i + \frac{1}{2}\Delta r}{\Delta r^2 r_i} u_{i+1,j,m}^{(\ell)} \\ + \frac{1}{\Delta z^2} \left[ u_{i,j-1,m}^{(\ell)} - 2u_{i,j,m}^{(\ell)} + u_{i,j+1,m}^{(\ell)} \right] + \frac{\lambda_m}{r_i^2} u_{i,j,m}^{(\ell)} = f_{i,j,m}^{(\ell)} \end{aligned} \quad (26)$$

for  $\ell = 1, 2; i = 2, 3, \dots, M; j = 2, 3, \dots, N$ ; where

$$\lambda_m = \frac{2}{\Delta \theta^2} \left[ \cos m\Delta\theta - 1 \right]$$

and

$$\begin{aligned} u_{i,1,m}^{(\ell)} &= u_{i,N+1,m}^{(\ell)} = 0 \\ u_{M+1,j,m}^{(\ell)} &= g_{j,m}^{(\ell)} \end{aligned} \quad (27)$$

Since  $u(0, \theta_k, z_j)$  and  $f(0, \theta_k, z_j)$  are constant as a function of  $\theta_k$ , from Eqs. (25) we find that

$$u_{1,j,m}^{(\ell)} = f_{1,j,m}^{(\ell)} = 0 \quad \text{for } m = 1, 2, \dots, L \quad (28)$$

and that  $u_{1,j,0}^{(1)}$  and  $f_{1,j,0}^{(1)}$  are just the unknown values of the solution and the right side at the point  $(0, 0, z_j)$ . To develop an equation for  $u_{1,j,0}^{(1)}$ , we substitute Eq. (25a) into Eq. (23) to get

$$\begin{aligned} \frac{4}{\Delta r^2} \left[ -u_{1,j,0}^{(1)} + u_{2,j,0}^{(1)} \right] + \frac{1}{\Delta z^2} \left[ u_{1,j-1,0}^{(1)} - 2u_{1,j,0}^{(1)} + u_{1,j+1,0}^{(1)} \right] \\ = f(0, 0, z_j). \end{aligned} \quad (29)$$

To solve the problem, then, for  $m = 1, 2, \dots, L$  we solve the Helmholtz Eq. (26) subject to the conditions of Eqs. (27) and (28) by calling PWSCYL and specifying that the solution is known on all four boundaries (MBDCND = 1, NBDCND = 1). For  $m = 0$ , we solve Eqs. (26) and (29) subject to the conditions of Eq. (27) by calling PWSCYL with the requirement that the solution be unspecified at  $r = 0$  but be specified on the other three boundaries (MBDCND = 5, NBDCND = 1).

After all the coefficients  $u_{i,j,m}^{(l)}$  have been computed, the approximation  $u_{i,j,k}$  is computed by Eq. (25a), using the FFT routine.

#### SAMPLE PROBLEM

In this section we illustrate the use of subroutine PWSCYL by presenting a FORTRAN program which solves the particular problem:

$$\frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial u}{\partial r} \right) + \frac{\partial^2 u}{\partial z^2} = (2rz)^2 (4z^2 + 3r^2)$$

on the rectangle  $0 < r < 1$ ,  $0 < z < 1$  with the boundary conditions

$$\left. \begin{array}{l} u(0,z) \text{ unspecified} \\ \frac{\partial u}{\partial r}(1,z) = 4z^4 \end{array} \right\} \quad 0 \leq z \leq 1$$

and

$$\left. \begin{array}{l} \frac{\partial u}{\partial z}(r,0) = 0 \\ \frac{\partial u}{\partial z}(r,1) = 4r^4 \end{array} \right\} \quad 0 \leq r \leq 1.$$

The solution to this problem is not unique. It is a one-parameter family of solutions given by

$$u_c(r,z) = (rz)^4 + c ,$$

where  $c$  is an arbitrary constant.

The  $r$ -interval  $[0,1]$  will be divided into 50 panels and the  $z$ -interval  $[0,1]$  into 100 panels. The finite difference approximation to this problem may be solved by the following program:

PROGRAM XAMPLE

C

C PROGRAM TO ILLUSTRATE THE USE OF PWSCYL.

C

DIMENSION F(75,105) ,BDA(101) ,BDB(101) ,  
1 BDC(51) ,BDD(51) ,W(1014) ,R(51) ,  
2 Z(101)

C

C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF. ALSO NOTE THAT W  
C IS DIMENSIONED  $6*(N+1) + 8*(M+1)$ .

C

IDIMF = 75  
INTL = 0  
A = 0.  
B = 1.  
M = 50  
MBDCND = 6  
C = 0.  
D = 1.  
N = 100  
NBDCND = 3  
ELMBDA = 0.

C

C AUXILIARY QUANTITIES.

C

MP1 = M+1  
NP1 = N+1

C

C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.

C

DO 100 I=1,MP1  
R(I) = (I-1)/50.  
100 CONTINUE  
DO 105 J=1,NP1  
Z(J) = (J-1)/100.  
105 CONTINUE

C

C GENERATE BOUNDARY DATA.

C

DO 110 J=1,NP1  
BDB(J) = 4.\*Z(J)\*\*4  
110 CONTINUE  
DO 115 I=1,MP1  
BDC(I) = 0.  
BDD(I) = 4.\*R(I)\*\*4  
115 CONTINUE

C

C BDA IS A DUMMY VARIABLE.

C

```
C  
C      GENERATE RIGHT SIDE OF EQUATION.  
C  
DO 125 I=1,MP1  
    DO 120 J=1,NP1  
        F(I,J) = 4.*R(I)**2*Z(J)**2*(4.*Z(J)**2 + 3.*R(I)**2)  
120    CONTINUE  
125 CONTINUE  
    CALL PWSCYL (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
1                 ELMBDA,F,IDLIMF,PERTRB,IERROR,W)  
C  
C      COMPUTE DISCRETIZATION ERROR BY MINIMIZING OVER ALL A THE FUNCTION  
C      NORM(F(I,J) - A*I - U(R(I),Z(J))). THE EXACT SOLUTION IS  
C      U(R,Z) = (R*Z)**4 + ARBITRARY CONSTANT.  
C  
X = 0.  
DO 135 I=1,MP1  
    DO 130 J=1,NP1  
        X = X+F(I,J) - (R(I)*Z(J))**4  
130    CONTINUE  
135 CONTINUE  
X = X/(NP1*MP1)  
DO 145 I=1,MP1  
    DO 140 J=1,NP1  
        F(I,J) = F(I,J)-X  
140    CONTINUE  
145 CONTINUE  
ERR = 0.  
DO 155 I=1,MP1  
    DO 150 J=1,NP1  
        X = ABS(F(I,J)-(R(I)*Z(J))**4  
        IF (X .GT. ERR) ERR = X  
150    CONTINUE  
155 CONTINUE  
PRINT 1001 , PERTRB,IERROR,ERR  
STOP  
C  
1001 FORMAT (///,9H PERTRB =,E12.5,4X,8HIERROR =,I2,4X,  
1           22HDISCRETIZATION ERROR =,E12.5)  
C  
END
```

After the CALL to PWSCYL, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed along with the output parameter PERTRB. Since there is not a unique solution, we took as our exact solution the function:

$$u_{\alpha}(r,z) = (rz) + \alpha$$

where  $\alpha$  is determined so that  $u_\alpha$  is the closest solution to the approximation  $v$  in the sense that  $\alpha$  minimizes

$$\sum_{i=1}^{M+1} \sum_{j=1}^{N+1} (v_{i,j} - u(r_i, z_j))^2 .$$

It can be shown that the proper  $\alpha$  is

$$\alpha = \frac{1}{(M+1)(N+1)} \sum_{i=1}^{M+1} \sum_{j=1}^{N+1} (v_{i,j} - (r_i z_j)^4) .$$

The output line is:

PERTRB = 2.26743E-04    IERROR = 0    DISCRETIZATION ERROR = 3.73672E-04

Note that even though the proper condition was satisfied for the continuous problem, the necessary condition for the approximation was not satisfied and, hence, the right side of the linear system of equations had to be perturbed by the constant PERTRB to guarantee a solution.

Also note that the discretization error indicates that the approximation has four digits of accuracy. This can be compared with the roundoff error reported in the Timing and Accuracy subsection above.

#### IV. SUBROUTINE PWSCSP

In this chapter we describe the subroutine PWSCSP which solves a finite difference approximation to the Helmholtz equation in spherical coordinates assuming axisymmetry (no dependence on longitude).

$$\frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \frac{\partial u}{\partial r}) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \frac{\partial u}{\partial \theta}) + \frac{\lambda}{r^2 \sin^2 \theta} u = f(\theta, r)$$

where  $\theta$  is colatitude and  $r$  is the radial coordinate.

##### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL PWSCSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF, RS, RF, N, NBDCND,  
            BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
```

where the arguments are defined as:

###### On Input

###### INTL

- = 0 on initial entry to PWSCSP or if any of the arguments RS, RF, N, NBDCND are changed from a previous call
- = 1 if RS, RF, N, NBDCND are all unchanged from previous call to PWSCSP

Note: A call with INTL = 0 takes approximately three times as much time as a call with INTL = 1. Once a call with INTL = 0 has been made then subsequent solutions corresponding to different F, BDTS, BDTF, BDRS, BDRF can be obtained faster with INTL = 1 since initialization is not repeated.

###### TS, TF

The range of  $\theta$  (colatitude); i.e.,  $TS \leq \theta \leq TF$ . TS must be less than TF. TS and TF are in radians. A TS of zero corresponds to the North Pole and a TF of  $\pi$  corresponds to the South Pole.

###### M

The number of panels into which the interval (TS,TF) is subdivided. Hence, there will be  $M + 1$  grid points in the  $\theta$ -direction given by  $\theta_I = (I-1)\Delta\theta + TS$  for  $I = 1, 2, \dots, M + 1$ , where  $\Delta\theta = (TF-TS)/M$  is the panel width.

MBDCND

Indicates the type of boundary condition at  $\theta = TS$  and  $\theta = TF$

- = 1 if the solution is specified at  $\theta = TS$  and  $\theta = TF$
- = 2 if the solution is specified at  $\theta = TS$  and the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TF$  (see note 2 below)
- = 3 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  and  $\theta = TF$  (see notes 1,2 below)
- = 4 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  (see note 1 below) and the solution is specified at  $\theta = TF$
- = 5 if the solution is unspecified at  $\theta = TS = 0$  and the solution is specified at  $\theta = TF$
- = 6 if the solution is unspecified at  $\theta = TS = 0$  and the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TF$  (see note 2 below)
- = 7 if the solution is specified at  $\theta = TS$  and the solution is unspecified at  $\theta = TF = \pi$
- = 8 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  (see note 1 below) and the solution is unspecified at  $\theta = TF = \pi$
- = 9 if the solution is unspecified at  $\theta = TS = 0$  and  $\theta = TF = \pi$

Notes: 1. If  $TS = 0$ , do not use  $MBDCND = 3, 4$ , or  $8$ , but instead use  $MBDCND = 5, 6$ , or  $9$ .  
2. If  $TF = \pi$ , do not use  $MBDCND = 2, 3$ , or  $6$ , but instead use  $MBDCND = 7, 8$ , or  $9$ .

BDTS

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = TS$ . When  $MBDCND = 3, 4$ , or  $8$ ,

$$BDTS(J) = \frac{\partial u}{\partial \theta}(TS, r_J) , \quad J = 1, 2, \dots, N+1 .$$

When  $MBDCND$  has any other value,  $BDTS$  is a dummy variable.

BDTF

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = TF$ . When MBDCND = 2, 3, or 6,

$$BDTF(J) = \frac{\partial u}{\partial \theta} (TF, r_J) , \quad J = 1, 2, \dots, N+1$$

When MBDCND has any other value, BDTF is a dummy variable.

RS, RF

The range of  $r$ ; i.e.,  $RS \leq r \leq RF$ . RS must be less than RF. RS must be non-negative.

N

The number of panels into which the interval (RS,RF) is subdivided. Hence, there will be  $N + 1$  grid points in the  $r$ -direction given by  $r_J = (J-1)\Delta r + RS$  for  $J = 1, 2, \dots, N+1$ , where  $\Delta r = (RF-RS)/N$  is the panel width. Let  $k$  be an integer greater than one, then  $N$  must have the following form depending on NBDCND (see below). If NBDCND = 2, 4, or 6,  $N$  must have the form  $2^k - 1$ . If NBDCND = 1 or 5,  $N$  must have the form  $2^k$ . If NBDCND = 3,  $N$  must have the form  $2^k - 2$ .

NBDCND

Indicates the type of boundary condition at  $r = RS$  and  $r = RF$ .

- = 1 if the solution is specified at  $r = RS$  and  $r = RF$ .
- = 2 if the solution is specified at  $r = RS$  and the derivative of the solution with respect to  $r$  is specified at  $r = RF$
- = 3 if the derivative of the solution with respect to  $r$  is specified at  $r = RS$  and  $r = RF$
- = 4 if the derivative of the solution with respect to  $r$  is specified at  $RS$  and the solution is specified at  $r = RF$
- = 5 if the solution is unspecified at  $r = RS = 0$  (see note below) and the solution is specified at  $r = RF$

= 6 if the solution is unspecified at  $r = RS = 0$   
(see note below) and the derivative of the solution  
with respect to  $r$  is specified at  $r = RF$

Note: NBDCND = 5 or 6 cannot be used with  
MBDCND = 1,2,4,5, or 7 (the former indicates  
that the solution is unspecified at  $r = 0$ ;  
the latter indicates that the solution  
is specified). Use instead NBDCND = 1 or 2.

#### BDRS

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = RS$ . When NBDCND = 3 or 4,

$$BDRS(I) = \frac{\partial u}{\partial r}(\theta_I, RS) , \quad I = 1, 2, \dots, M+1 .$$

When NBDCND has any other value, BDRS is a dummy variable.

#### BDRF

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution with respect to  $r$  at  $r = RF$ . When NBDCND = 2, 3, or 6,

$$BDRF(I) = \frac{\partial u}{\partial r}(\theta_I, RF) , \quad I = 1, 2, \dots, M+1 .$$

When NBDCND has any other value, BDRF is a dummy variable.

#### ELMUDA

The constant  $\lambda$  in the Helmholtz equation. If  $\lambda > 0$ , a solution may not exist. However, PWSCSP will attempt to find a solution. If NBDCND = 5 or 6 or MBDCND = 5,6,7,8, or 9, ELMUDA must be zero.

#### F

A two-dimensional array that specifies the value of the right side of the Helmholtz equation and boundary values (if any). For  $I = 2, 3, \dots, M$  and  $J = 2, 3, \dots, N$

$$F(I, J) = f(\theta_I, r_J) .$$

On the boundaries F is defined by

<u>MBDCND</u>	<u>F(I,J)</u>	<u>F(M+1)J)</u>
1	$u(TS, r_J)$	$u(TF, r_J)$
2	$u(TS, r_J)$	$f(TF, r_J)$
3	$f(TS, r_J)$	$f(TF, r_J)$
4	$f(TS, r_J)$	$u(TF, r_J)$
5	$f(0, r_J)$	$u(TF, r_J)$ $J = 1, 2, \dots, N + 1$
6	$f(0, r_J)$	$f(TF, r_J)$
7	$u(TS, r_J)$	$f(\pi, r_J)$
8	$f(TS, r_J)$	$f(\pi, r_J)$
9	$f(0, r_J)$	$f(\pi, r_J)$

<u>NBDCND</u>	<u>F(I,1)</u>	<u>F(I,N+1)</u>
1	$u(\theta_I, RS)$	$u(\theta_I, RF)$
2	$u(\theta_I, RS)$	$f(\theta_I, RF)$
3	$f(\theta_I, RS)$	$f(\theta_I, RF)$
4	$f(\theta_I, RS)$	$u(\theta_I, RF)$ $I = 1, 2, \dots, M + 1$
5	$f(TS, 0)$	$u(\theta_I, RF)$
6	$f(TS, 0)$	$f(\theta_I, RF)$

F must be dimensioned at least  $(M + 1) \times (N + 1)$ .

#### IDIMF

The row (or first) dimension of the array F as it appears in the program calling PWSCSP. This parameter is used to specify the variable dimension of F. IDIMF must be at least  $M + 1$ .

#### W

A one-dimensional array that must be provided by the user for work space. Let  $L = 2^k$  (see the definition of N). Then the length of W must be at least  $\left[ 2(L + 1)(k - 1) + 6(N + 1) + 12(M + 1) + 6 \right]$ .

On Output

F

Contains the solution  $U(I,J)$  of the finite difference approximation for the grid point  $(\theta_I, r_J)$ ,  $I = 1, 2, \dots, M + 1$ ,  $J = 1, 2, \dots, N + 1$ .

PERTRB

If a combination of periodic or derivative boundary conditions is specified for a Poisson equation ( $\lambda = 0$ ), a solution may not exist. PERTRB is a constant, calculated and subtracted from F, which ensures that a solution exists. PWSCSP then computes this solution, which is a least squares solution to the original approximation. This solution is not unique.

IERROR

An error flag that indicates invalid input parameters. Except for numbers 0 and 10, a solution is not attempted.

- = 0 no error
- = 1  $TS < 0$
- = 2  $TS \geq TF$
- = 3  $MBDCND < 1$  or  $MBDCND > 9$
- = 4  $RS < 0$
- = 5  $RS \geq RF$
- = 6  $NBDCND < 1$  or  $NBDCND > 6$
- = 7 N is not of the proper form
- = 8 An NBDCND of 5 or 6 is used with an MBDCND of 1, 2, 4, 5, or 7
- = 9 ELMBDA is non-zero and either MBDCND = 5, 6, 7, 8, or 9 or NBDCND = 5 or 6
- = 10 ELMBDA > 0
- = 11 IDIMF < M + 1

Since this is the only means of indicating a possibly incorrect call to PWSCSP, the user should test IERROR after a call.

W

Contains intermediate values that must not be destroyed if PWSCSP will be called again with INTL = 1.

Entry Points PWSCSP, PWSCS1

Special Conditions None

Common Blocks None

I/O None

Precision Single

Originator Paul N. Swarztrauber

Space Required 1353<sub>10</sub> locations on the NCAR Control Data 7600.

Accuracy and Timing The execution time is proportional to  $MN \log_2 N$  and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method, a uniform random number generator was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$ . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision, using the difference equations that correspond to the sample problem at the end of this chapter. With  $F(I,J)$  as a right side, subroutine PWSCSP was used to compute a solution  $U(I,J)$ . The maximum absolute difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR Control Data 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

M	N	Execution Time (msec)	Max. Absolute Difference
32	32	61	$2.1 \times 10^{-12}$
64	64	284	$1.7 \times 10^{-11}$
128	128	1341	$1.1 \times 10^{-10}$

Portability American National Standards Institute FORTRAN with no machine-dependent constants

Required Resident Routines SIN, COS, SQRT

#### DIFFERENCE APPROXIMATIONS

Subroutine PWSCSP sets up a finite difference approximation to the Helmholtz equation

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial u}{\partial r} \right) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} \left( \sin \theta \frac{\partial u}{\partial \theta} \right) + \frac{\lambda}{r^2 \sin^2 \theta} u = f(\theta, r) \quad (1)$$

on all or a portion of the cross section of the sphere defined by  $TS \leq \theta \leq TF$  and  $RS \leq r \leq RF$ . The subroutine also incorporates the given boundary data to form a linear system of equations, which is solved by subroutine BLKTRI (see Chapter VII).

To describe the system of equations formed, we determine a grid of points  $(\theta_i, r_j)$  by selecting integers M and N, which define grid spacings  $\Delta\theta = (TF-TS)/M$  and  $\Delta r = (RF-RS)/N$ , and the grid

$$\begin{aligned} \theta_i &= (i-1)\Delta\theta + TS & i &= 1, 2, \dots, M+1 \\ r_j &= (j-1)\Delta r + RS & j &= 1, 2, \dots, N+1 \end{aligned} \quad (2)$$

In order to determine an approximate solution  $u_{i,j} \approx u(\theta_i, r_j)$  to Eq. (1) we require that  $u_{i,j}$  satisfy the following finite difference approximation to Eq. (1)

$$\begin{aligned}
 & \frac{1}{r_j^2 \Delta r^2} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{i,j+1} - u_{i,j}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{i,j} - u_{i,j-1}) \right] \\
 & + \frac{1}{r_j^2 \Delta \theta^2 \sin \theta_i} \left[ \sin(\theta_i + \frac{1}{2}\Delta \theta) (u_{i+1,j} - u_{i,j}) \right. \\
 & \quad \left. - \sin(\theta_i - \frac{1}{2}\Delta \theta) (u_{i,j} - u_{i-1,j}) \right] \\
 & + \frac{\lambda}{r_j^2 \sin^2 \theta_i} u_{i,j} = f(\theta_i, r_j)
 \end{aligned} \tag{3}$$

This equation must be augmented by the equations that result from the boundary conditions and a special equation when the center of the sphere ( $r = 0$ ) is included. We will illustrate, by examples, the incorporation of boundary data into the linear equations. We will also develop the equations that are used at the singularities of the coordinate system.

### 1. Solution specified:

Assume that the solution is specified on the boundary  $\theta = TS$ ; i.e.,

$$u(TS, r) = g(r) \quad RS \leq r \leq RF$$

Then for  $j = 1, 2, \dots, N+1$ ,  $u_{1,j}$  is given by  $u_{1,j} = g(r_j)$ . Substituting into Eq. (3) evaluated for  $i = 2$ , we obtain

$$\begin{aligned}
 & \frac{1}{r_j^2 \Delta r^2} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{2,j+1} - u_{2,j}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{2,j} - u_{2,j-1}) \right] \\
 & + \frac{1}{r_j^2 \Delta \theta^2 \sin \theta_2} \left[ \sin(\theta_2 + \frac{1}{2}\Delta \theta) (u_{3,j} - u_{2,j}) - \sin(\theta_2 - \frac{1}{2}\Delta \theta) u_{2,j} \right] \\
 & + \frac{\lambda}{r_j^2 \sin^2 \theta_2} u_{2,j} = f(\theta_2, r_j) - \frac{\sin(\theta_2 - \frac{1}{2}\Delta \theta)}{r_j^2 \Delta \theta^2 \sin \theta_2} g(r_j)
 \end{aligned} \tag{4}$$

which illustrates how the boundary data are incorporated into the right side of the linear equations.

2. The derivative of the solution with respect to  $\theta$  specified ( $TS > 0$ ):

Assume that  $TS > 0$  and a function  $h(r)$  is specified such that

$$\frac{\partial u}{\partial \theta}(TS, r) = h(r) \quad (5)$$

Since we desire a second-order accurate approximation to the derivative at  $\theta = TS$ , we introduce a virtual point at  $\theta_0 = TS - \Delta\theta$  as a computational device. A finite difference approximation to Eq. (5) yields

$$u_{2,j} - u_{0,j} = 2\Delta\theta h(r_j) \quad u = 1, 2, \dots, N + 1 \quad (6)$$

The unknown  $u_{0,j}$  is eliminated by substituting Eq. (6) into Eq. (3) evaluated for  $i = 1$ .

$$\begin{aligned} & \frac{1}{r_j^2 \Delta r^2} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{1,j+1} - u_{1,j}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{1,j} - u_{1,j-1}) \right] \\ & + \frac{1}{r_j^2 \Delta \theta^2 \sin TS} \left\{ \left[ \sin(TS + \frac{1}{2}\Delta \theta) + (\sin(TS - \frac{1}{2}\Delta \theta)) \right] (u_{2,j} - u_{1,j}) \right\} \\ & + \frac{\lambda}{r_j^2 \sin^2 TS} u_{1,j} = f(TS, r_j) + \frac{2\sin(TS - \frac{1}{2}\Delta \theta)}{r_j^2 \Delta \theta \sin TS} h(r_j) \end{aligned} \quad (7)$$

As in condition 1 above, this illustrates how the boundary data are incorporated into the linear system. Note also that the form of the left side of the equation is altered from Eq. (3).

3. The solution unspecified at  $TS = 0$ :

Assume now that  $TS = 0$ ,  $RS > 0$  and the solution is unspecified. In this event, Eq. (3) cannot be used unless  $\lambda = 0$ . If in addition  $\frac{\partial u}{\partial \theta}(0, r) = 0$ , then L'Hospital's rule can be applied to the second term in Eq. (1) to obtain

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial u}{\partial r} \right) + \frac{2}{r^2} \frac{\partial^2 u}{\partial \theta^2} = f(0, r) \text{ at } \theta = 0 \quad . \quad (8)$$

The sample problem at the end of this chapter illustrates how, in practice, the conditions  $\lambda = 0$  and  $\frac{\partial u}{\partial \theta}(0, r) = 0$  are satisfied. As in condition 2

above we introduce a virtual point at  $\theta_0 = TS - \Delta\theta$ . Then the finite difference form of  $\frac{\partial u}{\partial \theta}(0, r) = 0$  is  $u_{0,j} = u_{2,j}$ . Substituting this into a finite difference approximation to Eq. (8), we obtain

$$\begin{aligned} & \frac{1}{r_j^2 \Delta r} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{1,j+1} - u_{1,j}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{1,j} - u_{1,j-1}) \right] \\ & + \frac{4}{r_j^2 \Delta \theta^2} (u_{2,j} - u_{1,j}) = f(0, r_j) \end{aligned} \quad (9)$$

as the equation that is used at  $\theta = TS = 0$ . A similar equation is used when  $\theta = TF = \pi$ .

#### 4. The solution unspecified at RS = 0:

Assume now that  $RS = 0$  so neither Eq. (3) nor Eq. (8) can be used. If we multiply Eq. (1) by  $r^2 \sin\theta$  and integrate over the sector  $0 \leq r \leq \frac{1}{2}\Delta r$  and  $TS \leq \theta \leq TF$ , we obtain

$$\begin{aligned} & \frac{1}{2}\Delta r^2 \int_{TS}^{TF} \sin\theta \frac{\partial u}{\partial r}(\theta, \frac{1}{2}\Delta r) d\theta + \int_0^{\frac{1}{2}\Delta r} \left[ \sin(TF) \frac{\partial u}{\partial \theta}(TF, r) \right. \\ & \left. - \sin(TS) \frac{\partial u}{\partial \theta}(TS, r) \right] dr = \int_0^{\frac{1}{2}\Delta r} r^2 \int_{TS}^{TF} \sin\theta f(\theta, r) dr d\theta \end{aligned} \quad (10)$$

Denote the first, second, and third integrals in Eq. (10) by  $I_1$ ,  $I_2$ , and  $I_3$ , respectively.

$$\begin{aligned} I_1 & \approx \frac{1}{4}\Delta r \Delta \theta \left[ w_1 (u_{1,2} - u_c) + w_2 (u_{M+1,2} - u_c) \right. \\ & \left. + \sum_{i=2}^M \sin(\theta_i) (u_{i,2} - u_c) \right], \end{aligned}$$

where  $u_c$  is the approximate solution at the center of the sphere  $r = 0$ . The weights  $w_1$  and  $w_2$  have the value  $\frac{1}{2}\sin\frac{1}{2}\Delta\theta/\cos\frac{1}{2}\Delta\theta$  if a derivative condition is specified.

If  $TS = 0$  then  $w_1 = \frac{1}{4}\sin^{\frac{1}{2}}\Delta\theta$ , if  $TF = \pi$  then  $w_2 = \frac{1}{4}\sin^{\frac{1}{2}}\Delta\theta$ . These weights are chosen so that the coefficient matrix of the linear system is self-adjoint under an inner product which yields a least squares solution via a constant perturbation of  $F$  (in the event that the system is singular). Reference [6] contains a description of how the least squares solution is obtained on the surface of the sphere.

Next, we consider  $I_2$ . If  $u(\theta, r)$  is smooth at  $r = 0$ , the limit, as  $r$  goes to zero, of  $\frac{\partial u}{\partial \theta}$  must be zero since

$$\frac{\partial u}{\partial \theta} = r(\cos\theta \frac{\partial u}{\partial x} - \sin\theta \frac{\partial u}{\partial z})$$

Hence, a linear approximation on the interval  $0 \leq r \leq \Delta r$  has the form

$$\frac{\partial u}{\partial \theta}(TS, r) \approx \frac{r}{\Delta r} \frac{\partial u}{\partial \theta}(TS, \Delta r)$$

$$\frac{\partial u}{\partial \theta}(TF, r) \approx \frac{r}{\Delta r} \frac{\partial u}{\partial \theta}(TF, \Delta r)$$

Substituting these into  $I_2$  we obtain

$$I_2 \approx \frac{\Delta r}{8} \left[ \sin(TF)BDTF(\Delta r) - \sin(TS)BDTS(\Delta r) \right]$$

In  $I_3$  we approximate  $f(\theta, r)$  by its value  $f_c$  at the center of the sphere to obtain

$$I_3 \approx \frac{\Delta r^3}{24} [\cos(TS) - \cos(TF)] f_c$$

Finally, substituting the approximate values for  $I_1$ ,  $I_2$ , and  $I_3$  and dividing by  $\frac{\Delta r^3}{24} [\cos(TS) - \cos(TF)]$ , we obtain the equation that is used at the center of the sphere

$$\begin{aligned}
 & \frac{6\Delta\theta}{\Delta r^2 [\cos(TS) - \cos(TF)]} \left[ w_1(u_{1,2} - u_c) + w_2(u_{M+1,2} - u_c) \right. \\
 & \quad \left. + \sum_{i=2}^M \sin(\theta_i)(u_{i,2} - u_c) \right] \\
 & = f_c - \frac{3}{\Delta r^2 [\cos(TS) - \cos(TF)]} \left[ \sin(TF)BDTF(\Delta r) \right. \\
 & \quad \left. - \sin(TS)BDTS(\Delta r) \right]. \tag{11}
 \end{aligned}$$

Subroutine PWSCSP sets up the linear equations and incorporates the boundary data as illustrated above. This system is solved by subroutine BLKTRI (see Chapter VII).

#### SINGULAR PROBLEMS

In the appendix it is shown that if a system is singular, certain restrictions are placed on the data that must be satisfied if a solution is to exist. Fundamental to this analysis is a vector  $h$ , which has a component corresponding to each unknown on a subrectangle of the region  $TS \leq \theta \leq TF$ ,  $RS \leq r \leq RF$ . Hence we can denote the components of  $h$  by  $h_{I,J}$  for  $I = IS, \dots, IF$  and  $J = JS, \dots, JF$  where  $IS$ ,  $IF$ ,  $JS$ , and  $JF$  depend on the boundary conditions and are defined below. If the center of the sphere is included,  $h$  has a component we denote by  $h_c$ . Further  $h_{I,J}$  has the form  $h_{I,J} = h\theta(I) \cdot hr(J)$ . The restriction  $h^T f = 0$  (given in the Appendix) then takes the form

$$h_c f_c + \sum_{I=IS}^{IF} \sum_{J=JS}^{JF} h\theta(I) \cdot hr(J) \cdot f_{I,J} = 0, \tag{12}$$

where  $f_c$  and  $f_{I,J}$  are the right side of the Helmholtz equation adjusted by the boundary data as in Eqs. (7) and (11) above, and  $f_c$  is the adjusted right side at the center of the sphere. The incorporation of the boundary data is also discussed in detail in Chapter VII, which describes subroutine BLKTRI. The quantities  $h\theta(I)$  and  $hr(J)$  are defined

in the tables that follow. If the values of MBDCND and NBDCND do not appear, the problem is not singular and the results of this section do not apply.

$h\theta(I)$

MBDCND	IS	$h\theta(IS)$	IF	$h\theta(IF)$
3	1	$\frac{\sin(TS + \frac{1}{2}\Delta\theta)}{2\cos\frac{1}{2}\Delta\theta}$	M+1	$\frac{\sin(TF - \frac{1}{2}\Delta\theta)}{2\cos\frac{1}{2}\Delta\theta}$
6	1	$\frac{1}{4}\sin\frac{1}{2}\Delta\theta$	M+1	$\frac{\sin(TF - \Delta\theta)}{2\cos\frac{1}{2}\Delta\theta}$
8	1	$\frac{\sin(TS + \frac{1}{2}\Delta\theta)}{2\cos\frac{1}{2}\Delta\theta}$	M+1	$\frac{1}{4}\sin\frac{1}{2}\Delta\theta$
9	1	$\frac{1}{4}\sin\frac{1}{2}\Delta\theta$	M+1	$\frac{1}{4}\sin\frac{1}{2}\Delta\theta$

For  $I = IS+1, \dots, IF - 1$ ,  $h\theta(I) = \sin\theta_I$ .

$hr(J)$

NBDCND	JS	$hr(JS)$	JF	$hr(JF)$
3	1	$RS^2 \frac{(RS + \frac{1}{2}\Delta r)^2}{2RS^2 + \frac{1}{2}\Delta r^2}$	N+1	$RF^2 \frac{(RF - \frac{1}{2}\Delta r)^2}{2RF^2 + \frac{1}{2}\Delta r^2}$
6	2	$r_2^2$	N+1	$RF^2 \frac{(RF - \frac{1}{2}\Delta r)^2}{2RF^2 + \frac{1}{2}\Delta r^2}$

For  $J = JS+1, \dots, JF - 1$ ,  $hr(J) = r_J^2$ .

The value of  $h_c$  is either  $\frac{\Delta r^2}{24\Delta\theta} (\cos TS - \cos TF)$  or 0, depending on whether or not the center of the sphere is included.

Usually, as a result of computational or observational errors,  $f_c$  and  $f_{I,J}$  will not satisfy Eq. (12), hence,  $h^T f \neq 0$  and a solution will not exist. However, if  $f$  is perturbed by computing a vector  $g = f - (h^T f / h^T e)e$ , where  $e$  is the vector whose components are all 1, then  $h^T g = 0$  and if  $f$  is replaced by  $g$ , a solution exists. Furthermore, it is shown in the Appendix that this solution is a least squares solution to the unperturbed system with right side  $f$ . The

solution, plus a constant, is also a solution; hence, the solution is not unique.

The constant

$$\text{PERTRB} = \frac{h^T f}{h^T e}$$

is computed by subroutine PWSCSP and is returned in the argument list. If the solution is checked by differencing and comparing with  $f$ , the results will differ by the constant PERTRB.

From Eq. (12) and the definition of  $e$ ,

$$h^T e = h_c^T + \sum_{I=IS}^{IF} \sum_{J=JS}^{JF} h\theta(I) \cdot h_r(J) . \quad (13)$$

### THREE-DIMENSIONAL PROBLEMS

In this section we will describe by example how to use PWSCSP to solve the three-dimensional Poisson equation on the interior of the sphere. It will be assumed that the solution is specified on the surface of the sphere. In addition, we will assume equatorial symmetry. Hence we wish to approximate a function  $u(\theta, r, \phi)$  which satisfies

$$\begin{aligned} & \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \frac{\partial u}{\partial r}) + \frac{1}{r^2 \sin \theta} \frac{\partial}{\partial \theta} (\sin \theta \frac{\partial u}{\partial \theta}) & r < r_0 \\ & + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2 u}{\partial \phi^2} = f(\theta, r, \phi) & 0 < \theta < \pi/2 \\ & & 0 < \phi < 2\pi \end{aligned} \quad (14)$$

subject to the boundary conditions

$$u(\theta, r_0, \phi) = g(\theta, \phi) \quad \text{at } r = r_0$$

$$\frac{\partial u}{\partial \theta}(0, r, \phi) = 0 \quad \text{equatorial symmetry} .$$

Define the grid spacings  $\Delta\theta = \pi/(2M)$ ,  $\Delta r = r_0/N$ , and  $\Delta\phi = 2\pi/L$  and the grid

$$\begin{aligned}\theta_i &= (i-1)\Delta\theta & i &= 1, \dots, M+1 \\ r_j &= (j-1)\Delta r & j &= 1, \dots, N+1 \\ \phi_k &= (k-1)\Delta\phi & k &= 1, \dots, L+1\end{aligned}$$

We require that the approximate solution  $u_{i,j,k} \approx u(\theta_i, r_j, \phi_k)$  satisfy the finite difference approximation

$$\begin{aligned}&\frac{1}{r_j^2 \Delta r^2} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{i,j+1,k} - u_{i,j,k}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{i,j,k} - u_{i,j-1,k}) \right] \\ &+ \frac{1}{r_j^2 \Delta \theta^2 \sin \theta_i} \cdot \left[ \sin(\theta_i + \frac{1}{2}\Delta\theta) (u_{i+1,j,k} - u_{i,j,k}) \right. \\ &\quad \left. - \sin(\theta_i - \frac{1}{2}\Delta\theta) (u_{i,j,k} - u_{i-1,j,k}) \right] \\ &+ \frac{1}{r_j^2 \Delta \phi^2 \sin^2 \theta_i} (u_{i,j,k+1} - 2u_{i,j,k} + u_{i,j,k-1}) = f_{i,j,k} \quad .\end{aligned}\tag{15}$$

For purposes of exposition we will assume that  $f_{i,j,k}$  has the form  $f_{i,j} \sin \omega k \Delta \phi$  and that  $g_{i,k}$  has the form  $g_i \sin \omega k \Delta \phi$ . The case of general  $f_{i,j,k}$  and  $g_{i,k}$  will be discussed at the end of this section. Note that  $f_{i,j}$  and  $g_i$  are just the Fourier coefficients of  $f_{i,j,k}$  and  $g_{i,k}$ . We seek a solution  $u_{i,j,k}$  of the form  $u_{i,j} \sin \omega k \Delta \phi$ . Substituting the forms of  $u_{i,j,k}$ ,  $f_{i,j,k}$ , and  $g_{i,k}$  into Eq. (15) and dividing by  $\sin \omega k \Delta \phi$ , we find that the Fourier coefficients must satisfy

$$\begin{aligned}&\frac{1}{r_j \Delta r^2} \left[ (r_j + \frac{1}{2}\Delta r)^2 (u_{i,j+1} - u_{i,j}) - (r_j - \frac{1}{2}\Delta r)^2 (u_{i,j} - u_{i,j-1}) \right] \\ &+ \frac{1}{r_j^2 \Delta \theta^2 \sin \theta_i} \left[ \sin(\theta_i + \frac{1}{2}\Delta\theta) (u_{i+1,j} - u_{i,j}) \right. \\ &\quad \left. - \sin(\theta_i - \frac{1}{2}\Delta\theta) (u_{i,j} - u_{i-1,j}) \right] - \frac{2(1-\cos \omega k \Delta \phi)}{r_j^2 \Delta \phi^2 \sin^2 \theta_i} u_{i,j} = f_{i,j} \quad ,\end{aligned}\tag{16}$$

and on the boundaries

$$u_{i,N+1} = g_i$$

and

$$u_{M+2,j} = u_{M,j}$$

Since Eq. (16) has the same form as Eq. (3), PWSCSP can be used to determine  $u_{i,j}$ , after which the solution is given by the Fourier synthesis  $u_{i,j,k} = u_{i,j} \sin \omega k \Delta \phi$ . For general  $f_{i,j,k}$  one first Fourier transforms in the  $\phi$ -direction to obtain the Fourier coefficients  $f_{i,j}^{(1)}(n)$  and  $f_{i,j}^{(2)}(n)$  in the expression

$$f_{i,j,k} = \sum_{n=0}^{L/2} f_{i,j}^{(1)}(n) \sin(nk\Delta\phi) + f_{i,j}^{(2)}(n) \cos(nk\Delta\phi)$$

Similarly, one obtains the Fourier coefficients  $g_i^{(1)}(n)$ ,  $g_i^{(2)}(n)$  in the expression

$$g_{i,k} = \sum_{n=0}^{L/2} g_i^{(1)}(n) \sin(nk\Delta\phi) + g_i^{(2)}(n) \cos(nk\Delta\phi) .$$

Then each of the Fourier coefficients  $u_{i,j}^{(1)}(n)$  and  $u_{i,j}^{(2)}(n)$  in the expression

$$u_{i,j,k} = \sum_{n=0}^{L/2} u_{i,j}^{(1)}(n) \sin(nk\Delta\phi) + u_{i,j}^{(2)}(n) \cos(nk\Delta\phi) \quad (17)$$

satisfies Eq. (16) and hence, PWSCSP can be called repeatedly (L times) to obtain them. Finally, a Fourier synthesis of Eq. (17) yields the solution.

#### SAMPLE PROBLEM

In this section, we present a FORTRAN program that illustrates the use of PWSCSP to solve the three-dimensional problem described in the previous section. In particular, we assume a  $2.5^\circ$  grid; hence,  $M = 36$ . We assume also that  $N = 16$  and  $r_0 = 1$ .

The first part of the program solves a problem that does not have longitudinal dependence ( $\omega = \lambda = 0$ ), namely

$$f(\theta, r, \phi) = 12r^2 \cos^2 \theta$$

$$g(\theta, \phi) = \cos^4 \theta \quad r_0 = 1$$

The solution to the continuous problem is then

$$u(\theta, r, \phi) = r^4 \cos^4 \theta$$

which is just  $z^4$  in Cartesian coordinates. Since there is no  $\phi$  dependence, the Fourier coefficients are just

$$f_{i,j} = 12r_j^2 \cos^2 \theta_i$$

$$g_i = \cos^4 \theta_i$$

In the second part of the program we assume

$$f(\theta, r, \phi) = 0$$

$$g(\theta, \phi) = \sin \theta \sin \phi$$

The solution to the continuous problem is

$$u(\theta, r, \phi) = r \sin \theta \sin \phi$$

which has longitudinal dependence. The Fourier coefficients are just

$$f_{i,j} = 0 \quad \text{and} \quad g_i = \sin \theta_i$$

and

$$\lambda = -2(1-\cos \Delta \phi) / \Delta \phi^2$$

In both cases the discrete solution is compared to the continuous solution.

```
C
C      PROGRAM TO ILLUSTRATE THE USE OF PWSCSP
C
C      DIMENSION      F(48,33)      ,BDTF(33)      ,W(956),R(33) ,THETA(48)
C
C      THE VALUE OF IDIMF IS THE FIRST DIMENSION OF F. SINCE N=31,L=32
C      THEREFORE K=5 AND W IS DIMENSIONED 2(L+1)(K-1)+6(N+1)+12(M+1)+6
C      = 956    (M=36)
```

```
C
PI = 3.14159265358979
INTL = 0
TS = 0.
TF = PI/2.
M = 36
MBDCND = 6
RS = 0.
RF = 1.
N = 32
NBDCND = 5
ELMBDA = 0.
IDIMF = 48
C
C      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING THE
C      BOUNDARY DATA AND THE RIGHT SIDE OF THE EQUATION.
C
MP1 = M+1
DTHETA = TF/M
DO 100 I=1,MP1
    THETA(I) = (I-1)*DTHETA
100 CONTINUE
NP1 = N+1
DR = 1./N
DO 105 J=1,NP1
    R(J) = (J-1)*DR
105 CONTINUE
C
C      GENERATE NORMAL DERIVATIVE DATA AT EQUATOR
C
DO 110 J=1,NP1
    BDTF(J) = 0.
110 CONTINUE
C
C      COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE
C
DO 115 I=1,MP1
    F(I,N+1) = COS(THETA(I))**4
115 CONTINUE
C
C      COMPUTE RIGHT SIDE OF EQUATION
C
DO 125 I = 1,MP1
    CI4 = 12.*COS(THETA(I))**2
    DO 120 J=1,N
        F(I,J) = CI4*R(J)**2
120     CONTINUE
125 CONTINUE
C
CALL PWSCSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF, RS, RF, N, NBDCND, BDRS,
            BDRF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
```

```
C
C      COMPUTE DISCRETIZATION ERROR
C
ERR = 0.
DO 135 I=1,MP1
    CI4 = COS(THETA(I))**4
    DO 130 J=1,N
        Z = ABS(F(I,J)-CI4*R(J)**4)
        IF (Z .GT. ERR) ERR = Z
130    CONTINUE
135    CONTINUE
    PRINT 1001 , IERROR,ERR
C
C      THE FOLLOWING PROGRAM ILLUSTRATES THE USE OF PSWCSP TO SOLVE
C      A THREE DIMENSIONAL PROBLEM WHICH HAS LONGITUDINAL DEPENDENCE
C
MBDCND = 2
NBDCND = 1
DPHI = PI/72
ELMBDA = -2.*(1-COS(DPHI))/DPHI**2
C
C      COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE
C
DO 140 I=1,MP1
    F(I,N+1) = SIN(THETA(I))
140    CONTINUE
C
C      COMPUTE RIGHT SIDE OF THE EQUATION
C
DO 150 J=1,N
    DO 145 I=1,MP1
        F(I,J) = 0.
145    CONTINUE
150    CONTINUE
C
CALL PWSCSP (INTL,TS,TF,M,MBDCND,BDTS,BDTF,RS,RF,N,NBDCND,BDRS,
1           BDRF,ELMBDA,F,IDLIMF,PERTRB,IERROR,W)
C
C      COMPUTE DISCRETIZATION ERROR      (FOURIER COEFFICIENTS)
C
ERR = 0
DO 160 I=1,MP1
    SI = SIN(THETA(I))
    DO 155 J=1,NP1
        Z = ABS(F(I,J)-R(J)*SI)
        IF (Z .GT. ERR) ERR = Z
155    CONTINUE
160    CONTINUE
```

```
C
PRINT 1001 , IERROR,ERR
STOP
C
1001 FORMAT (///9H IERROR = I3,10X,22HDISCRETIZATION ERROR = E12.4)
C
END
```

After the CALL to PWSCSP, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed. The output lines are:

```
IERROR = 0           DISCRETIZATION ERROR = 7.9984E-04
IERROR = 0           DISCRETIZATION ERROR = 5.8682E-05
```

## V. SUBROUTINE PWSSSP

In this chapter we describe the subroutine PWSSSP which solves a finite difference approximation to the Helmholtz equation in spherical coordinates and on the surface of the unit sphere (radius of 1).

$$\frac{1}{\sin\theta} \frac{\partial}{\partial\theta} (\sin\theta \frac{\partial u}{\partial\theta}) + \frac{1}{\sin^2\theta} \frac{\partial^2 u}{\partial\phi^2} + \lambda u = f(\theta, \phi)$$

where  $\theta$  is colatitude and  $\phi$  is longitude.

### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL PWSSSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF, PS, PF, N, NBDCND,  
             BDPS, BDPF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
```

where the arguments are defined as:

On Input

INTL

- = 0 on initial entry to PWSSSP or if N or NBDCND are changed from a previous call.
- = 1 if N and NBDCND are unchanged from a previous call

Note: A call with INTL = 1 is about 1% faster than a call with INTL = 0.

TS,TF

The range of  $\theta$  (colatitude); i.e.,  $TS < \theta < TF$ . TS must be less than TF. TS and TF are in radians. A TS of zero corresponds to the north pole and a TF of  $\pi$  corresponds to the south pole.

M

The number of panels into which the interval (TS,TF) is subdivided. Hence, there will be  $M + 1$  grid points in the  $\theta$ -direction given by  $\theta_I = (I-1)\Delta\theta + TS$  for  $I = 1, 2, \dots, M + 1$ , where  $\Delta\theta = (TF-TS)/M$  is the panel width.

MBDCND

Indicates the type of boundary condition at  $\theta = TS$  and  $\theta = TF$ .

- = 1 if the solution is specified at  $\theta = TS$  and  $\theta = TF$
- = 2 if the solution is specified at  $\theta = TS$  and the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TF$  (see note 2 below).
- = 3 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  and  $\theta = TF$  (see notes 1, 2 below).
- = 4 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  (see note 1 below) and the solution is specified at  $\theta = TF$ .
- = 5 if the solution is unspecified at  $\theta = TS = 0$  and the solution is specified at  $\theta = TF$ .
- = 6 if the solution is unspecified at  $\theta = TS = 0$  and the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TF$  (see note 2 below).
- = 7 if the solution is specified at  $\theta = TS$  and the solution is unspecified at  $\theta = TF = \pi$ .
- = 8 if the derivative of the solution with respect to  $\theta$  is specified at  $\theta = TS$  (see note 1 below) and the solution is unspecified at  $\theta = TF = \pi$ .
- = 9 if the solution is unspecified at  $\theta = TS = 0$  and  $\theta = TF = \pi$ .

Notes: 1. If  $TS = 0$ , do not use  $MBDCND = 3, 4$ , or  $8$ , but instead use  $MBDCND = 5, 6$ , or  $9$ .  
2. If  $TF = \pi$ , do not use  $MBDCND = 2, 3$ , or  $6$ , but instead use  $MBDCND = 7, 8$ , or  $9$ .

#### BDTS

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = TS$ . When  $MBDCND = 3, 4$ , or  $8$ ,

$$BDTS(J) = \frac{\partial u}{\partial \theta} (TS, \phi_J) \quad , \quad J = 1, 2, \dots, N + 1.$$

When  $MBDCND$  has any other value,  $BDTS$  is a dummy variable.

#### BDTF

A one-dimensional array of length  $N + 1$  that specifies the values of the derivative of the solution with respect to  $\theta$  at  $\theta = TF$ . When  $MBDCND = 2, 3$ , or  $6$ , then

$$BDTF(J) = \frac{\partial u}{\partial \theta}(TF, \phi_J) \quad , \quad J = 1, 2, \dots, N + 1$$

When MBDCND has any other value, BDTF is a dummy variable.

#### PS,PF

The range of  $\phi$  (longitude); i.e.,  $PS \leq \phi \leq PF$ . PS must be less than PF. PS and PF are in radians. If PS = 0 and PF =  $2\pi$ , periodic boundary conditions are usually prescribed.

#### N

The number of panels into which the interval (PS,PF) is subdivided. Hence, there will be  $N + 1$  grid points in the  $\phi$ -direction given by  $\phi_J = (J-1) \cdot \Delta\phi + PS$  for  $J = 1, 2, \dots, N + 1$ , where  $\Delta\phi = (PF - PS)/N$  is the panel width.  $N$  must be of the form  $2^p 3^q 5^r$  where  $p, q$ , and  $r$  are any non-negative integers.  $N$  must be greater than 2.

#### NBDCND

Indicates the type of boundary condition at  $\phi = PS$  and  $\phi = PF$ .

- = 0 if the solution is periodic in  $\phi$ ; i.e.,  $U(I,1) = U(I,N+1)$ .
- = 1 if the solution is specified at  $\phi = PS$  and  $\phi = PF$  (see note below).
- = 2 if the solution is specified at  $\phi = PS$  (see note below) and the derivative of the solution with respect to  $\phi$  is specified at  $\phi = PF$ .
- = 3 if the derivative of the solution with respect to  $\phi$  is specified at  $\phi = PS$  and  $\phi = PF$ .
- = 4 if the derivative of the solution with respect to  $\phi$  is specified at  $PS$  and the solution is specified at  $\phi = PF$  (see note below).

Note: NBDCND = 1, 2, or 4 cannot be used with MBDCND = 5, 6, 7, 8, or 9 (the former indicates that the solution is specified at a pole; the latter indicates that the solution is unspecified). Use instead MBDCND = 1 or 2.

#### BDPS

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution

with respect to  $\phi$  at  $\phi = PS$ . When NBDCND = 3 or 4,

$$BDPS(I) = \frac{\partial u}{\partial \phi}(\theta_I, PS), I = 1, 2, \dots, M + 1.$$

When NBDCND has any other value, BDPS is a dummy variable.

#### BDPF

A one-dimensional array of length  $M + 1$  that specifies the values of the derivative of the solution with respect to  $\phi$  at  $\phi = PF$ . When NBDCND = 2 or 3,

$$BDPF(I) = \frac{\partial u}{\partial \phi}(\theta_I, PF), I = 1, 2, \dots, M + 1.$$

When NBDCND has any other value, BDPF is a dummy variable.

#### ELMBDA

The constant  $\lambda$  in the Helmholtz equation. If  $\lambda > 0$ , a solution may not exist. However, PWSSSP will attempt to find a solution.

#### F

A two-dimensional array that specifies the value of the right side of the Helmholtz equation and boundary values (if any). For  $I = 2, 3, \dots, M$  and  $J = 2, 3, \dots, N$

$$F(I,J) = f(\theta_I, \phi_J).$$

On the boundaries F is defined by

<u>MBDCND</u>	<u><math>F(1,J)</math></u>	<u><math>F(M+1,J)</math></u>
1	$u(TS, \phi_J)$	$u(TF, \phi_J)$
2	$u(TS, \phi_J)$	$f(TF, \phi_J)$
3	$f(TS, \phi_J)$	$f(TF, \phi_J)$
4	$f(TS, \phi_J)$	$u(TF, \phi_J)$
5	$f(0, PS)$	$u(TF, \phi_J) J = 1, 2, \dots, N + 1$
6	$f(0, PS)$	$f(TF, \phi_J)$
7	$u(TS, \phi_J)$	$f(\pi, PS)$
8	$f(TS, \phi_J)$	$f(\pi, PS)$
9	$f(0, PS)$	$f(\pi, PS)$

<u>NBDCND</u>	<u>F(I,1)</u>	<u>F(I,N+1)</u>
0	$f(\theta_I, PS)$	$f(\theta_I, PS)$
1	$u(\theta_I, PS)$	$u(\theta_I, PF)$
2	$u(\theta_I, PS)$	$f(\theta_I, PF) \quad I = 1, 2, \dots, M + 1$
3	$f(\theta_I, PS)$	$f(\theta_I, PF)$
4	$f(\theta_I, PS)$	$u(\theta_I, PF)$

F must be dimensioned at least  $(M + 1) \times (N + 1)$ .

#### IDIMF

The row (or first) dimension of the array F as it appears in the program calling PWSSSP. This parameter is used to specify the variable dimension of F. IDIMF must be at least M + 1.

#### W

A one-dimensional array that must be provided by the user for work space. The length of W must be at least  $11(M + 1) + 6(N + 1)$ .

#### On Output

#### F

Contains the solution  $U(I,J)$  of the finite difference approximation for the grid point  $(\theta_I, \phi_J)$ ,  $I = 1, 2, \dots, M + 1$ ,  $J = 1, 2, \dots, N + 1$ .

#### PERTRB

If one specifies a combination of periodic, derivative or unspecified boundary conditions for a Poisson equation ( $\lambda=0$ ), a solution may not exist. PERTRB is a constant, calculated and subtracted from F, which ensures that a solution exists. PWSSSP then computes this solution, which is a least squares solution to the original approximation. This solution is not unique.

#### IERROR

An error flag that indicates invalid input parameters. Except for numbers 0 and 8, a solution is not attempted.

- = 0 no error
- = 1 TS < 0
- = 2 TS  $\geq$  TF
- = 3 MBDCND < 1 or MBDCND > 9

- = 4  $PS \geq PF$
- = 5 N is not of the form  $2^p 3^q 5^r$  or  $N \leq 2$
- = 6  $NBDCND < 0$  or  $NBDCND > 4$
- = 7 An NBDCND of 1, 2, or 4 is used with an MBDCND of 5, 6, 7, 8, or 9
- = 8  $ELMBDA > 0$
- = 9  $IDIMF < M + 1$

Since this is the only means of indicating a possibly incorrect call to PWSSSP, the user should test IERROR after a call.

W

Contains intermediate values that must not be destroyed if PWSSSP will be called again with INTL = 1 .

Entry Points PWSSSP, PWSS1, NCHECK

Special Conditions None

Common Blocks None

I/O None

Precision Single

Originator Paul N. Swarztrauber

Space Required 1432<sub>10</sub> locations on the NCAR Control Data 7600

Accuracy and Timing The execution time is proportional to  $MN \log_2 N$  and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method a uniform random number generator was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$  . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision, using the difference equations that correspond to the sample problem at the end of this chapter. With  $F(I,J)$  as a right side, subroutine PWSSSP was used to compute a solution  $U(I,J)$ . The maximum absolute

difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

N		Execution Time (msec)	Max. Absolute Difference
8	32	14	$2.3 \times 10^{-13}$
16	64	60	$1.3 \times 10^{-12}$
32	128	248	$7.7 \times 10^{-12}$

If  $NBDCND \neq 0$ , the execution times can be halved.

Portability American National Standards Institute FORTRAN with no machine-dependent constants.

Required  
Resident  
Routines SIN, COS

#### DIFFERENCE APPROXIMATION

Subroutine PWSSSP sets up a finite difference approximation to the Helmholtz equation

$$\frac{1}{\sin\theta} \frac{\partial}{\partial\theta} \left( \sin\theta \frac{\partial u}{\partial\theta} \right) + \frac{1}{\sin^2\theta} \frac{\partial^2 u}{\partial\phi^2} + \lambda u = f(\theta, \phi) \quad (1)$$

on all or a portion of the surface of the sphere defined by  $TS \leq \theta \leq TF$  and  $PS \leq \phi \leq PF$ . The subroutine also incorporates the given boundary data to form a linear system of equations which is solved by subroutine POIS (see Chapter VI).

To describe the system of equations formed, we determine a grid of points  $(\theta_i, \phi_j)$  by selecting integers M and N, which define grid spacings  $\Delta\theta = (TF-TS)/M$  and  $\Delta\phi = (PF-PS)/N$ , and the grid

$$\begin{aligned}\theta_i &= (i-1)\Delta\theta + TS & i = 1, 2, \dots, M+1 \\ \phi_j &= (j-1)\Delta\phi + PS & j = 1, 2, \dots, N+1\end{aligned}\quad (2)$$

In order to determine an approximate solution  $u_{i,j} \approx u(\theta_i, \phi_j)$  to Eq. (1), we require that  $u_{i,j}$  satisfy the following finite difference approximation to Eq. (1)

$$\begin{aligned}&\frac{1}{\Delta\theta^2 \sin\theta_i} \left[ \sin(\theta_i + \frac{1}{2}\Delta\theta)(u_{i+1,j} - u_{i,j}) \right. \\ &\quad \left. - \sin(\theta_i - \frac{1}{2}\Delta\theta)(u_{i,j} - u_{i-1,j}) \right] \\ &+ \frac{1}{(\Delta\phi \sin\theta_i)^2} (u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) + \lambda u_{i,j} \\ &= f(\theta_i, \phi_j)\end{aligned}\quad (3)$$

This equation must be augmented by the equations that result from the boundary conditions and special equations when the poles are included. We will illustrate, by examples, the incorporation of boundary data into the linear system of equations. We will also develop the equations that are used at the singularities of the coordinate systems (poles).

### 1. Solution specified:

Assume that the solution is specified on the boundary  $\theta = TS$ ; i.e.,

$$u(TS, \phi) = g(\phi) \quad PS \leq \phi \leq PF$$

Then for  $j = 1, 2, \dots, N+1$ ,  $u_{i,j}$  is given by  $u_{i,j} = g(\phi_j)$ . Substituting into Eq. (3) evaluated for  $i = 2$  we obtain

$$\begin{aligned}&\frac{1}{\Delta\theta^2 \sin\theta_2} \left[ \sin(\theta_2 + \frac{1}{2}\Delta\theta)(u_{3,j} - u_{2,j}) - \sin(\theta_2 - \frac{1}{2}\Delta\theta)u_{2,j} \right] \\ &+ \frac{1}{(\Delta\phi \sin\theta_2)^2} (u_{2,j+1} - 2u_{2,j} + u_{2,j-1}) + \lambda u_{2,j} \\ &= f(\theta_2, \phi_j) - \frac{\sin(\theta_2 - \frac{1}{2}\Delta\theta)}{\Delta\theta^2 \sin\theta_2} g(\phi_j)\end{aligned}\quad (4)$$

which illustrates how the boundary data are incorporated into the right

side of the linear equations.

2. The derivative of the solution with respect to  $\theta$  specified ( $TS > 0$ ):

Assume that  $TS > 0$  and a function  $h(\phi)$  is specified such that

$$\frac{\partial u}{\partial \theta} (TS, \phi) = h(\phi) . \quad (5)$$

Since we desire an approximation of second-order accuracy to the derivative at  $\theta = TS$ , we introduce a virtual point at  $\theta_0 = TS - \Delta\theta$  as a computational device. A finite difference approximation to Eq. (5) then yields

$$u_{2,j} - u_{0,j} = 2\Delta\theta h(\phi_j) \quad (6)$$

The unknown  $u_{0,j}$  is eliminated by substituting Eq. (6) into Eq. (3) evaluated for  $i = 1$

$$\begin{aligned} & \frac{1}{\Delta\theta^2 \sin TS} \left\{ \left[ \sin(TS + \frac{1}{2}\Delta\theta) + \sin(TS - \frac{1}{2}\Delta\theta) \right] (u_{2,j} - u_{1,j}) \right\} \\ & + \frac{1}{(\Delta\phi \sin TS)^2} (u_{1,j+1} - 2u_{1,j} + u_{1,j-1}) + \lambda u_{1,j} \\ & = f(TS, \phi_j) + \frac{2\sin(TS - \frac{1}{2}\Delta\theta)}{\Delta\theta \sin TS} h(\phi_j). \end{aligned} \quad (7)$$

As in condition 1 above, this illustrates how the boundary data are incorporated into the linear system. Note also that the form of the left side of this equation is altered from Eq. (3).

3. The solution periodic in  $\phi$ :

In this event  $u(\theta, \phi) = u(\theta, \phi + PF)$  and the approximate solution satisfies  $u_{i,j} = u_{i,j+N}$  where  $j + N$  is evaluated modulo  $N$ . The linear system of equations consists of Eq. (3) evaluated for  $i = 1, 2, \dots, N$  with the unknowns  $u_{0,j}$  and  $u_{N+1,j}$  replaced by  $u_{N,j}$  and  $u_{1,j}$ , respectively. The form of the system is significantly altered and the system of equations is solved by an algorithm which varies from the other cases.

When periodicity is assumed, it is not necessary to have  $PF - PS = 2\pi$ . However, it is likely that  $k(PF - PS) = 2\pi$  for some

integer  $k$  that would correspond to the wave number under consideration. This permits the user to determine solutions for high wave numbers with less computation.

4. The solution unspecified at  $TS = 0$  (north pole):

Assume that  $TS = 0$  and hence Eq. (3) cannot be used. Instead, we multiply Eq. (1) by  $\sin\theta$  and integrate over the spherical segment  $0 \leq \theta \leq \frac{1}{2}\Delta\theta$ ,  $PS \leq \phi \leq PF$  to obtain

$$\begin{aligned} & \sin \frac{1}{2}\Delta\theta \int_{PS}^{PF} \frac{\partial u}{\partial \theta}(\frac{1}{2}\Delta\theta, \phi) d\phi + \int_0^{\frac{1}{2}\Delta\theta} \frac{1}{\sin\theta} \left[ \frac{\partial u}{\partial \phi}(\theta, PF) - \frac{\partial u}{\partial \phi}(\theta, PS) \right] d\theta \\ & + \lambda \int_{PS}^{PF} \int_0^{\frac{1}{2}\Delta\theta} \sin\theta u(\theta, \phi) d\theta d\phi = \int_{PS}^{PF} \int_0^{\frac{1}{2}\Delta\theta} \sin\theta f(\theta, \phi) d\theta d\phi \end{aligned} \quad (8)$$

or

$$I_1 + I_2 + I_3 = I_4$$

First we approximate

$$\frac{\partial u}{\partial \theta}(\frac{1}{2}\Delta\theta, \phi) \approx \frac{u(\Delta\theta, \phi) - u(0, 0)}{\Delta\theta} \text{ and } \sin \frac{1}{2}\Delta\theta \approx \frac{1}{2}\Delta\theta \quad (9)$$

and apply the trapezoidal rule to obtain

$$I_1 = \frac{1}{2}\Delta\phi \left[ \sum_{J=2}^N u(\Delta\theta, \phi_J) + \frac{1}{2}u(\Delta\theta, PS) + \frac{1}{2}u(\Delta\theta, PF) - Nu(0, 0) \right] \quad (10)$$

If  $u(\theta, \phi)$  is smooth at the poles,

$\lim_{\theta \rightarrow 0} \frac{\partial u}{\partial \phi} = 0$  since  $u(\theta, \phi)$  as a function of  $\phi$  approaches a constant value, namely  $u(0, \phi)$  at the pole. Hence, a linear approximation on the interval  $0 \leq \theta \leq \Delta\theta$  has the form:

$$\frac{\partial u}{\partial \phi}(\theta, PS) \approx \frac{\theta}{\Delta\theta} \frac{\partial u}{\partial \phi}(\Delta\theta, PS) = \frac{\theta}{\Delta\theta} BDPS(2) \quad (11)$$

$$\frac{\partial u}{\partial \phi}(\theta, PF) \approx \frac{\theta}{\Delta\theta} \frac{\partial u}{\partial \phi}(\Delta\theta, PF) = \frac{\theta}{\Delta\theta} BDPF(2)$$

If, in addition, we assume  $\sin\theta \approx \theta$ ,

$$I_2 \approx \frac{1}{2} [BDPF(\Delta\theta) - BDPS(\Delta\theta)] \quad . \quad (12)$$

If we approximate  $u(\theta, \phi)$  by  $u(0,0)$ , we obtain

$$I_3 = \frac{N\Delta\phi\Delta\theta^2}{8} \lambda u(0,0) \quad (13)$$

and similarly if we approximate  $f(\theta, \phi)$  by  $f(0,0)$ ,

$$I_4 = \frac{N\Delta\phi\Delta\theta^2}{8} f(0,0) \quad . \quad (14)$$

Finally, we substitute into Eq. (8) the approximate values of the integrals and multiply through by  $8/(N\cdot\Delta\theta^2\Delta\phi)$  to obtain the formula that is used at the pole by PWSSSP.

$$\begin{aligned} & \frac{4}{N\cdot\Delta\theta^2} \left[ \frac{1}{2}u(\Delta\theta, PS) + \frac{1}{2}u(\Delta\theta, PF) + \sum_{J=2}^N u(\Delta\theta, \phi_J) - N \cdot u(0,0) \right] \\ & + \lambda u(0,0) = f(0,0) - \frac{4}{N\Delta\phi\Delta\theta^2} [BDPF(\Delta\theta) - BDPS(\Delta\theta)] \end{aligned} \quad (15)$$

A similar formula can be derived at the south pole.

Subroutine PWSSSP sets up the linear equations and incorporates the boundary data as illustrated above. This system is solved by subroutine POIS (see Chapter VI).

### SINGULAR PROBLEMS

In the Appendix it is shown that if a system is singular, certain restrictions are placed on the data that must be satisfied if a solution is to exist. Fundamental to this analysis is a vector  $h$ , which has a component corresponding to each unknown on a subrectangle of the region  $TS \leq \theta \leq TF$ ,  $PS \leq \phi \leq PF$ . Hence, we can denote the components of  $h$  by  $h_{I,J}$  for  $I = IS, \dots, IF$  and  $J = JS, \dots, JF$ , where  $IS$ ,  $IF$ ,  $JS$ , and  $JF$  depend on the boundary conditions and are defined below. If the poles are included, then  $h$  has components  $h_N$  and  $h_S$  which correspond to the north and south poles, respectively. Further,  $h_{I,J}$  has the form  $h_{I,J}^T = h_\theta(I) \cdot h_\phi(J)$ . The restriction  $h^T f = 0$  (given in the Appendix) then takes the form

$$h_N f_N + h_S f_S + \sum_{I=IS}^{IF} \sum_{J=JS}^{JF} h\theta(I) \cdot h\phi(J) \cdot f_{I,J} = 0 \quad (16)$$

where  $f_N$ ,  $f_S$  and  $f_{I,J}$  are the right side of the Helmholtz equation adjusted by the boundary data (if any) as in Eqs. (7) and (15), and  $f_N$  and  $f_S$  correspond to the north and south poles, respectively.

The incorporation of the boundary data is also discussed in detail in Chapter VII, which describes subroutine BLKTRI. The quantities  $h\theta(I)$  and  $h\phi(J)$  are defined in the tables that follow. If the values of MBDCND and NBDCND do not appear, the problem is not singular and the results of this section do not apply.

$h\theta(I)$

MBDCND	IS	$h\theta(IS)$	IF	$h\theta(IF)$
3	1	$\frac{\sin(TS+\frac{1}{2}\Delta\theta)}{2\cos^2\frac{1}{2}\Delta\theta}$	M+1	$\frac{\sin(TF-\frac{1}{2}\Delta\theta)}{2\cos^2\frac{1}{2}\Delta\theta}$
6	2	$\sin\Delta\theta$	M+1	$\frac{\sin(TF-\frac{1}{2}\Delta\theta)}{2\cos^2\frac{1}{2}\Delta\theta}$
8	1	$\frac{\sin(TS+\Delta\theta)}{2\cos^2\frac{1}{2}\Delta\theta}$	M	$\sin\Delta\theta$
9	2	$\sin\Delta\theta$	M	$\sin\Delta\theta$

For  $I = IS + 1, \dots, IF - 1$ ,  $h\theta(I) = \sin\theta_I$ .

The quantities  $h_N$  and  $h_S$  are either  $\frac{1}{4}N \sin \frac{1}{2}\Delta\theta$  or 0, depending on whether or not the corresponding pole is included.

$h\phi(J)$

NBDCND	JS	$h\phi(JS)$	JF	$h\phi(JF)$
0	1	1.	N	1.
3	1	.5	N+1	.5

For  $J = JS + 1, \dots, JF - 1$ ,  $h\phi(J) = 1$ .

Usually, as a result of computational or observational errors,  $f_N$ ,  $f_S$ , and  $f_{I,J}$  will not satisfy Eq. (16); hence,  $h^T f \neq 0$  and a solution will not exist. However, if  $f$  is perturbed by computing a vector  $g = f - (h^T f / h^T e) e$ , where  $e$  is the vector whose components are all 1, then  $h^T g = 0$  and if  $f$  is replaced by  $g$ , a solution exists. Furthermore, it is shown in the Appendix that this solution is a least squares solution to the unperturbed system with right side  $f$ . The solution plus a constant is also a solution; hence, the solution is not unique.

The constant

$$\text{PERTRB} = \frac{h^T f}{h^T e}$$

is computed by subroutine PWSSSP and is returned in the argument list. If the solution is checked by differencing and comparing with  $f$ , the results will differ by the constant PERTRB.

From Eq. (16) and the definition of  $e$ , we obtain

$$h^T e = h_N + h_S + \sum_{I=IS}^{IF} \sum_{J=JS}^{JF} h\theta(I)h\phi(J) . \quad (17)$$

The following identity is included for computing the sum in Eq. (17)

$$\sum_{I=IS}^{IF} \sin(I\Delta\theta) = \frac{\cos(IS-.5)\Delta\theta - \cos(IF+.5)\Delta\theta}{2\sin^{1/2}\Delta\theta} .$$

#### SAMPLE PROBLEM

In this section we present a FORTRAN program which illustrates the use of subroutine PWSSSP. The problem consists of solving Poisson's equation on the northern hemisphere subject to equatorial symmetry; i.e., the derivative of the solution at  $\theta = \pi/2$  is zero. A  $5^\circ$  grid is chosen

and, hence,  $M = 18$  and  $N = 72$ . The right side is selected to be  $F(\theta, \phi) = 2. - 6 \sin^2\theta \sin^2\phi$ , for which the exact solution is  $u(\theta, \phi) = \sin^2\theta \sin^2\phi$ .

Note in the output that PERTRB is not zero even though the continuous problem has a solution. The reason is that discretization errors result in a system which is not consistent.

```
C
C      PROGRAM TO ILLUSTRATE THE USE OF PWSSSP
C
C      DIMENSION      F(19,73)      ,BDTF(73)      ,SINT(19)
C      1            SINP(73)      ,W(647)
C
C      THE VALUE OF IDIMF IS THE FIRST DIMENSION OF F.  W IS DIMENSIONED
C      11*(M+1)+6X(N+1)=647 SINCE M=18 AND N=72
C
C      PI = 3.14159265358979
C      INTL = 0
C      TS = 0
C      TF = PI/2.
C      M = 18
C      MBDCND = 6
C      PS = 0
C      PF = PI+PI
C      N = 72
C      NBDCND = 0
C      ELMBDA = 0.
C      IDIMF = 19
C
C      GENERATE SINES FOR USE IN SUBSEQUENT COMPUTATIONS
C
C      DTHETA = TF/M
C      MP1 = M+1
C      DO 100 I=1,MP1
C          SINT(I) = SIN((I-1)*DTHETA)
C 100 CONTINUE
C      DPHI = (PI+PI)/N
C      NP1 = N+1
C      DO 105 J=1,NP1
C          SINP(J) = SIN((J-1)*DPHI)
C 105 CONTINUE
C
C      COMPUTE RIGHT SIDE OF EQUATION AND STORE IN F
C
C      DO 115 J=1,NP1
C          DO 110 I=1,MP1
C              F(I,J) = 2. - 6.*(SINT(I)*SINP(J))**2
C 110      CONTINUE
C 115      CONTINUE
```

```
C
C      STORE DERIVATIVE DATA AT THE EQUATOR
C
C      DO 120 J=1,NP1
C             BDTF(J) = 0.
120 CONTINUE
C
C      CALL PWSSSP (INTL,TS,TF,M,MBDCND,BDTS,BDTF,PS,PF,N,NBDCND,BDPS,
C      1           BDPF,ELMBDA,F,IDIMF,PERTRB,IERROR,W)
C
C      COMPUTE DISCRETIZATION ERROR. SINCE PROBLEM IS SINGULAR, THE
C      SOLUTION MUST BE NORMALIZED.
C
C      ERR = 0
C      DO 130 J=1,NP1
C             DO 125 I=1,MP1
C                   Z = ABS(F(I,J)-(SINT(I)*SINP(J))**2-F(1,1))
C                   IF (Z .GT. ERR) ERR = Z
125     CONTINUE
130     CONTINUE
C
C      PRINT 1001 , IERROR,ERR,PERTRB
C      STOP
C
C      1001 FORMAT (//9H IERROR= I3, 2X,22H DISCRETIZATION ERR = E12.4,
C      1           14H PERTURBATION=E12.4)
C
C      END
```

After the CALL to PWSSSP, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed along with the output parameter PERTRB. The printed output is:

IERROR= 0 DISCRETIZATION ERR = 3.3811E-03 PERTURBATION= 6.3220E-04

## VI. SUBROUTINE POIS

In this chapter we describe the subroutine POIS which solves the linear system of equations

$$A(I)*X(I-1,J) + B(I)*X(I,J) + C(I)*X(I+1,J) \quad I = 1, 2, \dots, M \\ + X(I,J-1) - 2.*X(I,J) + X(I,J+1) = Y(I,J) \quad J = 1, 2, \dots, N$$

The indices  $I \pm 1$  are evaluated modulo  $M$ ; i.e.,  $X(0,J) = X(M,J)$  and  $X(M+1,J) = X(1,J)$ , and  $X(I,0)$  may be equal to 0,  $X(I,2)$ , or  $X(I,N)$  and  $X(I,N+1)$  may be equal to 0,  $X(I,N-1)$ , or  $X(I,1)$  depending on an input parameter.

### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

The routine is used by the statement

```
CALL POIS (IFLG,NPEROD,N,MPEROD,M,A,B,C,IDLIM,Y,IERROR,W)
```

where the arguments are defined as:

#### On Input

##### IFLG

- = 0 on initial entry to POIS or if N and NPEROD are changed from previous call.
- = 1 if N and NPEROD are unchanged from previous call to POIS.

Note: A call with IFLG = 1 is about 1% faster than a call with IFLG = 0.

##### NPEROD

Indicates the values which  $X(I,0)$  and  $X(I,N+1)$  are assumed to have.

- = 0 if  $X(I,0) = X(I,N)$  and  $X(I,N+1) = X(I,1)$ .
- = 1 if  $X(I,0) = X(I,N+1) = 0$ .
- = 2 if  $X(I,0) = 0$  and  $X(I,N+1) = X(I,N-1)$ .
- = 3 if  $X(I,0) = X(I,2)$  and  $X(I,N+1) = X(I,N-1)$ .

##### N

The number of unknowns in the J-direction. N is dependent on NPEROD and must have the following form:

NPEROD

= 0 or 2, then  $N = 2^p 3^q 5^r$

= 1, then  $N = 2^p 3^q 5^r - 1$

= 3, then  $N = 2^p 3^q 5^r + 1$

where p, q, and r may be any non-negative integers.

N must be greater than 2.

MPEROD

= 0 if  $A(1)$  and  $A(M)$  are not zero

= 1 if  $A(1) = A(M) = 0$

M

The number of unknowns in the I-direction. M may be any integer greater than 1.

A,B,C

One-dimensional arrays of length M which specify the coefficients in the linear equations given above.

IDIMY

The row (or first) dimension of the two-dimensional array Y as it appears in the program calling POIS. This parameter is used to specify the variable dimension of Y. IDIMY must be at least M.

Y

A two-dimensional array which specifies the values of the right side of the linear system of equations given above. Y must be dimensioned at least  $M \times N$ .

W

A one-dimensional array which must be provided by the user for work space. The length of W must be at least  $6N + 5M$ .

On Output

Y

Contains the solution X.

IERROR

An error flag which indicates invalid input parameters. Except for number zero, a solution is not attempted.

= 0 no error  
= 1  $M \leq 2$   
= 2  $N \leq 2$  or not of the right form  
= 3  $IDIMY < M$   
= 4  $NPEROD < 0$  or  $NPEROD > 3$

W

Contains intermediate values that must not be destroyed if POIS will be called again with IFLG = 1.

Entry Points POIS, POISGN, POINIT, TRIDP, TRID

Special Conditions None

Common Blocks None

I/O None

Precision Single

Originator Roland Sweet

Space Required 305<sub>10</sub> locations on the NCAR Control Data 7600

Timing and Accuracy The execution time is proportional to  $MN \log N$  and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method a uniform random number generator was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$ . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision, using the difference equations that correspond to the sample problem at the end of this chapter. With  $F(I,J)$  as a right side, subroutine POIS was used to compute a solution  $U(I,J)$ . The maximum absolute difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error given in the sample problem.

M	N	Execution Time (msec)	Max. Absolute Difference
32	32	56	$5.8 \times 10^{-13}$
64	64	223	$2.9 \times 10^{-12}$
128	128	999	$1.4 \times 10^{-11}$

The execution times should be doubled if MPEROD = 0. They should be increased 30% if NPEROD = 3 and decreased 30% for any other value of NPEROD.

Portability

American National Standards Institute FORTRAN with no machine-dependent constants.

Required

COS

Resident

Routines

APPLICATION TO DIFFERENCE APPROXIMATIONS

Subroutine POIS may be used to solve a finite difference approximation to certain elliptic partial differential equations. Suppose we want to solve the equation

$$r(x) \frac{\partial^2 u}{\partial x^2} + s(x) \frac{\partial u}{\partial x} + t(x)u + \frac{\partial^2 u}{\partial y^2} = f(x, y) , \quad (1)$$

on the rectangle,  $a < x < b$ ,  $c < y < d$ , with certain boundary conditions imposed upon the solution  $u$  along the four boundaries. We define a grid with spacings  $\Delta x$  and  $\Delta y$  and approximate all derivatives of Eq. (1) by finite differences to obtain the equation

$$\begin{aligned} \frac{r_i}{\Delta x^2} \left[ v_{i-1,j} - 2v_{i,j} + v_{i+1,j} \right] + \frac{s_i}{2\Delta x} \left[ v_{i+1,j} - v_{i-1,j} \right] + t_i v_{i,j} \\ \frac{1}{\Delta y^2} \left[ v_{i,j-1} - 2v_{i,j} + v_{i,j+1} \right] = f_{i,j} , \end{aligned} \quad (2)$$

where  $r_i = r(x_i)$ ,  $s_i = s(x_i)$ ,  $t_i = t(x_i)$ ,  $f_{i,j} = f(x_i, y_j)$ , and  $v_{i,j}$  is an approximation to  $u(x_i, y_j)$ . This is the basic form of the equation which is used to solve for the unknowns  $v_{i,j}$ ,  $i = 1, 2, \dots, M$ ,  $j = 1, 2, \dots, N$ . The above equation must be modified at the boundaries to

reflect the effect of the given boundary conditions. By defining

$$a_i = r_i \frac{\Delta y^2}{\Delta x^2} - s_i \frac{\Delta y^2}{2\Delta x}$$

$$b_i = -2r_i \frac{\Delta y^2}{\Delta x^2} + t_i \Delta y^2$$

$$c_i = r_i \frac{\Delta y^2}{\Delta x^2} + s_i \frac{\Delta y^2}{2\Delta x}$$

$$y_{i,j} = \Delta y^2 f_{i,j}$$

$$Qv_{i,j} = v_{i,j-1} - 2v_{i,j} + v_{i,j+1}$$

Eq. (2) may be rewritten in the form

$$a_i v_{i-1,j} + b_i v_{i,j} + c_i v_{i+1,j} + Qv_{i,j} = y_{i,j} . \quad (3)$$

Two sets of boundary conditions are specified: one for the  $i$ -direction and one for the  $j$ -direction. We will discuss in detail the effect of boundary conditions in the  $i$ -direction.

#### 1. Solution specified - mixed boundary condition:

Assume that the boundary conditions are

$$u(a, y_j) = p_j \quad (4)$$

and

$$\alpha u(b, y_j) + \frac{\partial u}{\partial x}(b, y_j) = q_j , \quad (5)$$

where  $\alpha$  is a constant. Note that if  $\alpha = 0$ , we have the derivative boundary condition  $\frac{\partial u}{\partial x}(b, y_j) = q_j$ . For this case, we define  $\Delta x = \frac{b-a}{M}$  and

$$x_i = a + i\Delta x, \quad i = 0, 1, 2, \dots, M .$$

We see that the unknowns for a given  $y_j$  are  $v_{i,j}$  for  $i = 1, 2, \dots, M$ . For  $i = 2, 3, \dots, M-1$ , the unknowns satisfy (Eq.) (3). For  $i = 1$ , Eqs. (3) and (4) combine to

$$b_1 v_{1,j} + c_1 v_{2,j} + Qv_{1,j} = y_{1,j} - a_1 p_j \quad (6)$$

For  $i = M$ , we first approximate the boundary condition (5) by

$$\alpha v_{M,j} + \frac{v_{M+1,j} - v_{M-1,j}}{2\Delta x} = q_j \quad . \quad (7)$$

We solve this for  $v_{M+1,j}$  and substitute it for  $v_{M+1,j}$  in Eq. (3) to get

$$\begin{aligned} & (a_M + c_M) v_{M-1,j} + (b_M - 2\alpha \Delta x c_M) v_{M,j} + Qv_{M,j} \\ & = y_{M,j} - 2\Delta x c_M q_j \quad . \end{aligned} \quad (8)$$

From Eqs. (3), (6), and (8) we determine the quantities for input to POIS as:<sup>†</sup>

$$MPEROD = 1$$

$$A(1) = 0$$

$$B(1) = b_i$$

$$C(1) = c_i$$

$$Y(1,J) = y_{1,j} - a_1 p_1, \quad J=j=2,3,\dots,N-1$$

$$\left. \begin{array}{l} A(I) = a_i \\ B(I) = b_i \\ C(I) = c_i \\ Y(I,J) = y_{i,j}, \quad J=j=2,3,\dots,N-1 \end{array} \right\} I=i=2,3,\dots,M-1 \quad (9)$$

---

<sup>†</sup> Capital letters such as A, B, C, and Y refer to FORTRAN input arrays. Small letters such as  $a_i$ ,  $b_i$ ,  $c_i$ , and  $y_{i,j}$  refer to algebraic quantities.

$$A(M) = a_M + c_M$$

$$B(M) = b_M - 2\alpha \Delta x c_M$$

$$C(M) = 0$$

$$Y(M,J) = y_{M,j} - 2\Delta x c_M q_j , J = j = 2, 3, \dots, N-1$$

See the discussion below for the values of  $Y(I,J)$  at corners.

2. Mixed boundary condition - solution specified:

Suppose that the boundary conditions are

$$\alpha u(a, y_j) + \frac{\partial u}{\partial x}(a, y_j) = p_j \quad (10)$$

and

$$u(b, y_j) = q_j . \quad (11)$$

For this case we define  $\Delta x = \frac{b-a}{M}$  and

$$x_i = a + (i-1)\Delta x, \quad i = 1, 2, 3, \dots, M+1 .$$

We see that the unknowns for a given  $y_j$  in this case are still  $v_{i,j}$  for  $i = 1, 2, \dots, M$ . (However, note that in this case the subscript  $i$  refers to a different  $x$  value than in the previous case.)

For  $i = 2, 3, \dots, M-1$ , the unknowns satisfy Eq. (3). For  $i = 1$ , we discretize Eq. (10) to get

$$\alpha v_{1,j} + \frac{v_{2,j} - v_{0,j}}{2\Delta x} = p_j \quad (12)$$

and solve it for  $v_{0,j}$ . Substituting this value for  $v_{0,j}$  in Eq. (3) yields

$$\begin{aligned} & (b_1 + 2\Delta x \alpha a_1) v_{1,j} + (c_1 + a_1) v_{2,j} + Q v_{1,j} \\ & = y_{1,j} + 2\Delta x a_1 p_j . \end{aligned} \quad (13)$$

For  $i = M$ , Eq. (3) becomes

$$a_M v_{M-1,j} + b_M v_{M,j} + Q v_{M,j} = y_{M,j} - c_M q_j . \quad (14)$$

From Eqs. (3), (13), and (14) we determine the quantities for input to POIS as

$$\text{MPEROD} = 1$$

$$A(1) = 0$$

$$B(1) = b_1 + 2\Delta x \alpha a_1$$

$$C(1) = c_1 + a_1$$

$$Y(1,J) = y_{1,j} + 2\Delta x a_1 p_j , J = j = 2,3,\dots,N-1$$

$$A(M) = a_M$$

$$B(M) = b_M$$

$$C(M) = 0$$

$$Y(M,J) = y_{M,j} = c_M q_j , J = j = 2,3,\dots,N-1$$

and the quantities in Eq. (9).

See the discussion below for the values of  $Y(I,J)$  at corners.

### 3. Periodic solution specified:

Suppose now that the boundary conditions are given as

$$u(b+x, y_j) = u(a+x, y_j) \text{ for all } x. \quad (15)$$

For this case we define  $\Delta x = (b-a)/M$  and

$$x_i = a + (i-1)x , \quad i = 0,1,2,\dots,M+1$$

For a given  $y_j$  the unknowns  $v_{i,j}$ ,  $i = 2,3,\dots,M-1$ , satisfy Eq. (3). From Eq. (15) we get that

$$v_{0,j} = v_{M,j} \quad \text{and} \quad v_{M+1,j} = v_{1,j} .$$

For  $i = 1$ , Eq. (3) becomes

$$a_1 v_{M,j} + b_1 v_{1,j} + c_1 v_{2,j} + Q v_{1,j} = y_{1,j} \quad (16)$$

and for  $i = M$ , it becomes

$$a_M v_{M-1,j} + b_M v_{M,j} + c_M v_{1,j} + Q v_{M,j} = y_{M,j} \quad (17)$$

From Eqs. (3), (16), and (17) we determine the quantities for input to POIS as

$$MPEROD = 0$$

$$\left. \begin{array}{l} A(I) = a_i \\ B(I) = b_i \\ C(I) = c_i \\ Y(I,J) = y_{i,j}, J = j = 1, 2, \dots, N \end{array} \right\} I = i = 1, 2, 3, \dots, M$$

For the boundary conditions in the y-direction, the modifications to Eq. (3) proceed just as above for the x-direction. In this case there are no modifications necessary to the arrays  $A(I)$ ,  $B(I)$ , and  $C(I)$ , only to columns 1 and  $N$  of the input array  $Y$ . The proper form of  $Qv_{i,j}$  near the boundaries is determined in the program from the input parameter  $NPEROD$ .

For the y-direction, only solution-specified, derivative-specified, or periodic boundary conditions are permitted. Furthermore, although solution-specified at  $y = c$  and derivative-specified at  $y = d$  are allowed, the reverse is not. This should present no problem however, since a simple change of variables in the y-direction corrects this restriction.

The user must be careful at the corners to ensure that boundary data (if present) enter from both boundaries. In general this will affect the values of  $Y(1,1)$ ,  $Y(1,N)$ ,  $Y(M,1)$ , and  $Y(M,N)$ . For instance, suppose that the solution is specified at  $x(x_0) = a$  and  $y(y_0) = c$ . In this case we have  $u(a,y_j) = p_j$  and  $u(x_i,c) = q_i$ . For  $i = j = 1$  Eq. (3) becomes

$$a_1 v_{0,1} + b_1 v_{1,1} + c_1 v_{2,1} + v_{1,0} - 2v_{1,1} + v_{1,2} = y_{1,1} .$$

Using the boundary data we get

$$b_1 v_{1,1} + c_1 v_{2,1} - 2v_{1,1} + v_{1,2} = y_{1,1} - a_1 p_1 - q_1 .$$

Hence,  $Y(1,1)$  should be defined as

$$Y(1,1) = y_{1,1} - a_1 p_1 - q_1 .$$

Similar treatments are necessary for  $Y(1,N)$ ,  $Y(M,1)$ , and  $Y(M,N)$ .

DESCRIPTION OF ALGORITHM

Subroutine POIS solves the linear system of equations

$$a_i x_{i-1,j} + b_i x_{i,j} + c_i x_{i+1,j} + x_{i,j-1} - 2x_{i,j} + x_{i,j+1} = y_{i,j}, \quad (18)$$

$$i = 1, 2, \dots, M \quad j = 1, 2, \dots, N$$

where  $x_{0,j} = x_{M,j}$ ,  $x_{M+1,j} = x_{1,j}$

and  $x_{i,0} = \begin{cases} 0 & , \text{ or} \\ x_{i,2} & , \text{ or} \\ x_{i,N} \end{cases}$ ,  $x_{i,N+1} = \begin{cases} 0 & , \text{ or} \\ x_{i,N-1} & , \text{ or} \\ x_{i,1} \end{cases}$ .

Such systems of equations usually result from finite difference approximations to elliptic partial differential equations containing a term such as  $\frac{\partial^2 u}{\partial y^2}$ .

The system of equations is solved by the Buneman variant of cyclic reduction. To describe cyclic reduction we will consider the particular system composed of Eq. (18) with the conditions

$$a_1 = c_M = 0 \quad (19a)$$

$$x_{i,0} = x_{i,N+1} = 0, \quad i = 1, 2, \dots, M. \quad (19b)$$

We define the vectors

$$v_j = \begin{bmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{M,j} \end{bmatrix} \quad \text{and} \quad f_j = \begin{bmatrix} y_{1,j} \\ y_{2,j} \\ \vdots \\ y_{M,j} \end{bmatrix}, \quad j=1, 2, \dots, N,$$

and from Eq. (19b) note that

$$v_0 = v_{N+1} = 0 \quad . \quad (20)$$

Defining the  $M \times M$  matrix

$$A = \begin{bmatrix} b_1^{-2} & c_1 & & & \\ a_2 & b_2^{-2} & c_2 & & \circ \\ \cdot & \cdot & \cdot & & \\ \circ & a_{M-1} & b_{M-1}^{-2} & c_{M-1} & \\ & a_M & b_M^{-2} & & \end{bmatrix}$$

we can rewrite Eq. (18) as the system of matrix equations

$$v_{j-1} + Av_j + v_{j+1} = f_j \quad , \quad j = 1, 2, \dots, N \quad . \quad (22)$$

Suppose now that  $N$  has the special form

$$N = 2^{k+1} - 1 \quad . \quad (23)$$

Let us write down three consecutive Eqs. (22)

$$v_{j-2} + Av_{j-1} + v_j = f_{j-1} \quad (24a)$$

$$v_{j-1} + Av_j + v_{j+1} = f_j \quad (24b)$$

$$v_j + Av_{j+1} + v_{j+2} = f_{j+1} \quad . \quad (24c)$$

If we multiply Eq. (24b) by  $-A$  and add to it Eqs. (24a) and (24c), we arrive at the equation

$$v_{j-2} + A^{(1)} v_j + v_{j+2} = f_{j-1} + f_{j+1} - Af_j \quad , \quad (25)$$

where

$$A^{(1)} = 2I - A^2 \quad (26)$$

which involves only every other unknown. If we perform such linear combinations for  $j = 2, 4, \dots, N-1$ , we arrive at the new linear system

$$v_{j-2} + A^{(1)} v_j + v_{j+2} = f_j^{(1)} \quad , \quad j = 2, 4, \dots, N-1 \quad , \quad (27)$$

where

$$f_j^{(1)} = f_{j-1} + f_{j+1} - Af_j .$$

Notice that this is a system of equations in only the unknowns  $v_2, v_4, \dots, v_{N-1}$ , hence, the number of unknowns has been halved to  $2^k - 1$ . Furthermore, the system of Eq. (27) has exactly the same form as the original system of Eq. (22), so we can perform the same type of reduction again. In fact, we can repeatedly reduce the system of equations to half the size. At the  $r$ th step of this process ( $r = 0$  is the original system) we would have the system

$$v_{j-2^r} + A^{(r)} v_j + v_{j+2^r} = f_j^{(r)} , \quad j = 2^r, 2 \cdot 2^r, 3 \cdot 2^r, \dots, N+1-2^r \quad (28)$$

where

$$A^{(r)} = 2I - (A^{(r-1)})^2 \quad (29)$$

$$f_j^{(r)} = f_{j-2^{r-1}}^{(r-1)} + f_{j+2^{r-1}}^{(r-1)} - A^{(r-1)} f_j^{(r-1)} , \quad (30)$$

which has only the  $2^{k+1-r}-1$  unknowns  $v_{j+2^r}, j=1, 2, \dots, 2^{k+1-r}-1$ .

If we continue this reduction process for  $k$  steps we arrive at the single equation

$$A^{(k)} v_{2^k} = f_{2^k}^{(k)} . \quad (31)$$

In principle, this single equation may be readily solved by observing that the matrix  $A^{(k)}$  is a polynomial of degree  $2^k$  in the matrix  $A$ . This is seen from Eqs. (26) and (29). If we put

$$A^{(r)} = p_r(A)$$

where  $p_r$  is a polynomial of degree  $2^r$ , we find that the polynomials  $p_r(x)$  satisfy

$$p_0(x) = x$$

$$p_r(x) = 2 - [p_{r-1}(x)]^2 , \quad r = 1, 2, \dots, k .$$

Setting  $x = 2 \cos \theta$  we find that

$$p_r(2 \cos \theta) = 2 \cos 2^r \theta \quad , \quad r = 0, 1, \dots, k .$$

Hence, the roots of  $p_r$  are

$$x_j^{(r)} = 2 \cos \left[ \frac{(2j-1)\pi}{2^{r+1}} \right] \quad , \quad j = 1, 2, \dots, 2^r \quad ,$$

and so

$$A^{(r)} = \prod_{j=1}^{2^r} (A - x_j^{(r)} I) \quad . \quad (32)$$

Using Eq. (32), Eq. (31) becomes

$$\prod_{j=1}^{2^k} (A - x_j^{(k)} I) v_{2^k} = f_{2^k}^{(k)}$$

which is solved by solving the  $2^k$  systems

$$z_0 = \frac{f_{2^k}^{(k)}}{2^k}$$

$$(A - x_j^{(k)} I) z_j = z_{j-1} \quad , \quad j = 1, 2, \dots, 2^k$$

yielding in the end  $z_{2^k} = v_{2^k}$

Once  $v_{2^k}$  has been found, Eq. (28) with  $r = k - 1$  and the conditions of Eq. (20) yield

$$A^{(k-1)} v_{2^{k-1}} = \frac{f_{2^{k-1}}^{(k-1)}}{2^{k-1}} - v_{2^k}$$

$$A^{(k-1)} v_{3 \cdot 2^{k-1}} = \frac{f_{3 \cdot 2^{k-1}}^{(k-1)}}{3 \cdot 2^{k-1}} - v_{2^k} \quad ,$$

which may be solved for  $v_{2^{k-1}}$  and  $v_{3 \cdot 2^{k-1}}$  using the identity in Eq. (32).

Using Eq. (28) successively for  $r = k - 2, k - 3, \dots, 0$ , we can solve for the remaining unknowns. This process is known as cyclic reduction.

Unfortunately, the computation of the right sides  $f_j^{(r)}$  given by Eq. (30) is a numerically unstable process and suffers from a severe loss of significance even for relatively small values of  $r$ . To overcome this instability, Buneman expressed the vectors  $f_j^{(r)}$  as

$$f_j^{(r)} = A(r)p_j^{(r)} + q_j^{(r)} \quad (33)$$

and developed recurrence relations for the two sets of vectors  $p_j^{(r)}$  and  $q_j^{(r)}$ . Initially, one can write

$$f_j^{(0)} = f_j = A \cdot 0 + f_j$$

so we take  $p_j^{(0)} = 0$  and  $q_j^{(0)} = f_j$ . Then, substituting Eq. (33) into Eq. (30), we arrive at the relations

$$\left. \begin{aligned} p_j^{(r)} &= p_j^{(r-1)} + \left\{ A^{(r-1)} \right\}^{-1} \left[ q_j^{(r-1)} + p_{j-2}^{(r-1)} + p_{j+2}^{(r-1)} \right] \\ q_j^{(r)} &= q_{j-2}^{(r-1)} + q_{j+2}^{(r-1)} - 2p_j^{(r)} \end{aligned} \right\} \quad (34)$$

With  $f_j^{(r)}$  in the form of Eq. (33) with  $p_j^{(r)}$  and  $q_j^{(r)}$  defined by Eq. (34), the system of equations may be solved by cyclic reduction in a stable manner.

A complete description of the above algorithm may be found in reference [2] for the case  $N = 2^{k+1} - 1$ . For the generalization to a composite  $N$ , see reference [10].

### SAMPLE PROBLEM

In this section, we illustrate the use of subroutine POIS by presenting a FORTRAN program which solves the particular problem:

$$(1+x)^2 \frac{\partial^2 u}{\partial x^2} - 2(1+x) \frac{\partial u}{\partial x} + \frac{\partial^2 u}{\partial y^2} = 3(1+x)^4 \sin y \quad (35)$$

on the rectangle  $0 < x < 1$ ,  $-\pi < y < \pi$  with the boundary conditions:

$$\left. \begin{aligned} \frac{\partial u}{\partial x}(0, y) &= 4 \sin y \\ u(1, y) &= 16 \sin y \end{aligned} \right\} \quad -\pi \leq y \leq \pi \quad (36)$$

and with  $u$  periodic in  $y$ . (38)

The exact solution to this problem is:

$$u(x,y) = (1+x)^4 \sin y .$$

For the finite difference approximation, we subdivide the x-interval  $[0,1]$  into 20 panels by defining the grid points:

$$x_i = (i-1)\Delta x , \quad i = 1, 2, \dots, 21$$

where  $\Delta x = 1/20$ . Numbering the grid points in this fashion gives index 1 to the first unknown in the x-direction and index 20 to the last unknown; i.e.,  $M = 20$ . We subdivide the y-interval  $[-\pi, \pi]$  into 121 panels by defining grid points

$$y_j = -\pi + (j-1)\Delta y, \quad j = 1, 2, \dots, 121$$

where  $y = \pi/60$ . Numbering the grid points in this fashion gives index 1 to the first unknown in the y-direction and index 120 to the last unknown; i.e.,  $N = 120$ .

We now develop the difference equations which approximate Eq. (35). Let  $v_{ij}$  be the approximation to  $u(x_i, y_j)$ . Then using standard central difference approximations at the interior grid point  $(x_i, y_j)$ , we get:

$$\begin{aligned} & \frac{(1+x_i)^2}{\Delta x^2} (v_{i-1,j} - 2v_{i,j} + v_{i+1,j}) - \frac{2(1+x_i)}{2\Delta x} (v_{i+1,j} - v_{i-1,j}) \\ & + \frac{1}{\Delta y^2} (v_{i,j-1} - 2v_{i,j} + v_{i,j+1}) = 3(1+x_i)^4 \sin y_j , \end{aligned}$$

and multiplying by  $\Delta y^2$ , we get:

$$\begin{aligned} & [(1+x_i)^2 s^2 + (1+x_i)s^2 \Delta x] v_{i-1,j} - 2(1+x_i)^2 s^2 v_{ij} \\ & + [(1+x_i)^2 s^2 - (1+x_i)s^2 \Delta x] v_{i+1,j} + v_{i,j-1} - 2v_{i,j} + v_{i,j+1} \\ & = 3(1+x_i)^4 \Delta y^2 \sin y_j , \end{aligned} \tag{39}$$

where  $s = \Delta y / \Delta x$ .

From this equation we find that for  $i = 2, 3, \dots, 19$

$$\begin{aligned} A_i &= (1+x_i)^2 s^2 + (1+x_i) s^2 \Delta x \\ B_i &= -2(1+x_i)^2 s^2 \\ C_i &= (1+x_i)^2 s^2 - (1+x_i) s^2 \Delta x \\ f_{i,j} &= 3(1+x_i)^4 \Delta y^2 \sin y_j, \quad j = 1, 2, \dots, 120. \end{aligned}$$

(To avoid confusion with the grid points  $y_j$ , we call the right side of the equation  $f_{i,j}$  instead of  $y_{i,j}$ .) For  $i = 1$ , we must incorporate the boundary condition of Eq. (36) into Eq. (39). This is done by replacing Eq. (36) with the approximation:

$$\frac{1}{2\Delta x}(v_{2,j} - v_{0,j}) = 4 \sin y_j.$$

Incorporating this into Eq. (39), we obtain:

$$\begin{aligned} -2s^2 v_{1,j} + 2s^2 v_{2,j} + v_{1,j-1} - 2v_{1,j} + v_{1,j+1} \\ = (11 + 8/\Delta x) \Delta y^2 \sin y_j. \end{aligned}$$

Thus:

$$\begin{aligned} A_1 &= 0, \quad B_1 = -2s^2, \quad C_1 = 2s^2 \\ f_{1,j} &= (11 + 8/\Delta x) \Delta y^2 \sin y_j, \quad j = 1, 2, \dots, 120. \end{aligned}$$

For  $i = 20$ , we incorporate the boundary condition of Eq. (37) into Eq. (39) by putting

$$v_{20,j} = 16 \sin y_j$$

and substituting into Eq. (39) to get

$$[(1+x_{20})^2 s^2 + (1+x_{20}) s^2 \Delta x] v_{19,j} - 2(1+x_{20})^2 s^2 v_{20,j}$$

$$+ v_{20,j-1} - 2v_{20,j} + v_{20,j+1} = [3(1+x_{20})^4 - \frac{16}{\Delta x^2} (1+x_{20})^2 + \frac{16}{\Delta x} (1+x_{20})] \Delta y^2 \sin y_j.$$

From this equation we get

$$A_{20} = (1+x_{20})^2 s^2 + (1+x_{20}) s^2 \Delta x$$

$$B_{20} = -2(1+x_{20})^2 s^2$$

$$C_{20} = 0$$

$$f_{20,j} = [3(1+x_{20})^2 - \frac{16}{\Delta x^2} (1+x_{20})^2 + \frac{16}{\Delta x} (1+x_{20})] \Delta y^2 \sin y_j , j = 1, 2, \dots, 120.$$

The following program implements the solution:

```
PROGRAM XAMPLE
C
C PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE POIS.
C
C DIMENSION      F(25,130)      ,A(20)      ,B(20)      ,W(820),X(20)      ,
C 1                  Y(120)
C
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMY. ALSO NOTE THAT W
C IS DIMENSIONED 6*(N+1) + 5*(M+1).
C
C IDIMY = 25
C IFLG = 0
C MPEROD = 1
C M = 20
C DELTAX = 1./M
C NPEROD = 0
C N = 120
C PI = 3.14159265358979
C DELTAY = 2.*PI/N
C
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING
C COEFFICIENTS AND RIGHT SIDE OF EQUATION.
C
C DO 100 I=1,M
C     X(I) = (I-1)*DELTAX
100 CONTINUE
DO 105 J=1,120
Y(J) = -PI + (J-1)*DELTAY
105 CONTINUE
C
C GENERATE COEFFICIENTS.
```

```

C
S = (DELTAY/DELTAX)**2
T = S*DELTAX
A(1) = 0.
B(1) = -2.*S
C(1) = 2.*S
DO 110 I=2,M
    A(I) = (1.+X(I))**2*S + (1.+X(I))*T
    C(I) = (1.+X(I))**2*S - (1.+X(I))*T
    B(I) = -2.*(1.+X(I))**2*S
110 CONTINUE
    C(M) = 0.

C
C      GENERATE RIGHT SIDE OF EQUATION FOR I = 1 SHOWING INTRODUCTION OF
C      BOUNDARY DATA.
C
DYSQ = DELTAY**2
DO 115 J=1,N
    F(1,J) = DYSQ*(11. + 8./DELTAX)*SIN(Y(J))
115 CONTINUE
C
C      GENERATE RIGHT SIDE.
C
MM1 = M-1
DO 125 I=2,MM1
    DO 120 J=1,N
        F(I,J) = DYSQ*3.*(1.+X(I))**4*SIN(Y(J))
120     CONTINUE
125 CONTINUE
C
C      GENERATE RIGHT SIDE FOR I = M SHOWING INTRODUCTION OF
C      BOUNDARY DATA.
C
DO 130 J=1,N
    F(M,J) = DYSQ*(3.*(1+X(M))**4 - 16.*((1+X(M))/DELTAX)**2 +
1                               16.*((1+X(M))/DELTAX)*SIN(Y(J)))
130 CONTINUE
CALL POIS (IFLG,NPEROD,N,MPEROD,M,A,B,C,IDLIMY,F,IERROR,W)
C
C      COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS
C      U(X,Y) = (1+X)**4*SIN(Y)
C
ERR = 0.
DO 140 I=1,M
    DO 135 J=1,N
        Z = ABS(F(I,J)-(1.+X(I))**4*SIN(Y(J)))
        IF (Z .GT. ERR) ERR = Z
135     CONTINUE
140 CONTINUE
PRINT 1001 , IERROR,ERR
STOP

```

```
C  
1001 FORMAT (///,9H TERROR =,I3,10X,22HDISCRETIZATION ERROR =,E12.4)  
C  
END
```

After the CALL to POIS, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed. The output is:

```
IERROR = 0           DISCRETIZATION ERROR = 7.94113E-03
```

We see from the output that the approximation has almost three digits of accuracy. Note, however, the contrast between the discretization error and the roundoff error reported in the Timing and Accuracy subsection above.

## VII. SUBROUTINE BLKTRI

In this chapter we describe the subroutine BLKTRI which solves a system of linear equations of the form

$$AN(J)*X(I,J-1) + AM(I)*X(I-1,J) + (BN(J)+BM(I))*X(I,J)$$

$$+CN(J)*X(I,J+1) + CM(I)*X(I+1,J) = Y(I,J)$$

for  $I = 1, 2, \dots, M$  and  $J = 1, 2, \dots, N$ .

$I \pm 1$  is evaluated modulo  $M$  and  $J \pm 1$  modulo  $N$ ; i.e.,

$$X(I,0) = X(I,N); X(I,N+1) = X(I,1); X(0,J) = X(M,J);$$

$$X(M+1,J) = X(1,J).$$

These equations usually result from the discretization of separable elliptic equations. Boundary conditions may be Dirichlet, Neumann, or periodic.

### PARAMETER DESCRIPTION AND PROGRAM SPECIFICATIONS

This routine is used by the statement

```
CALL BLKTRI (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDLIM,Y,IERROR,W)
```

where the arguments are defined as

On Input            IFLG

= 0 initialization only. Certain quantities that depend on NP, N, AN, BN, and CN are computed and stored in the work array W.

= 1 the quantities that were computed in the initialization are used to obtain the solution  $X(I,J)$ .

Note: A call with IFLG = 0 takes approximately twice as much time as a call with IFLG = 1. However, the initialization does not have to be repeated unless NP, N, AN, BN, or CN change.

NP

= 0 if AN(1) and CN(N) are not zero, which corresponds to periodic boundary conditions.

= 1 if AN(1) and CN(N) are zero

N

The number of unknowns in the J-direction. If NP = 1, N must be of the form  $2^k - 1$  where k is an integer  $> 1$ .

If NP = 0, N must be of the form  $2^k$ . (The operation count of the algorithm is proportional to  $MN \log_2 N$  and, hence, N should be selected less than or equal to M.)

AN, BN, and CN

One-dimensional arrays of length N that specify the coefficients in the linear equations given above.

MP

= 0 if AM(1) and CM(M) are not zero, which corresponds to periodic boundary conditions  
= 1 if AM(1) = CM(M) = 0.

M

The number of unknowns in the I-direction. M may be any integer greater than 1.

AM, BM, and CM

One-dimensional arrays of length M that specify the coefficients in the linear equations given above.

IDIMY

The row (or first) dimension of the two-dimensional array Y as it appears in the program calling BLKTRI. This parameter is used to specify the variable dimension of Y. IDIMY must be at least M.

Y

A two-dimensional array that specifies the values of the right side of the linear system of equations given above. Y must be dimensioned at least  $M \times N$ .

W

A one-dimensional array that must be provided by the user for work space. If NP = 0, the length of W must be at least  $[2N \log_2 N + 2 + N + \max(2N, 6M)]$ . If NP = 1, the length of W must be at least  $[2(N + 2)(\log_2(N + 1) - 1) + 6M + 2]$ .

<u>On Output</u>	<p>Y Contains the solution X</p>
	<p>TERROR An error flag that indicates invalid input parameters. Except for number zero, a solution is not attempted.</p>
	<p>= 0 No error = 1 <math>M &lt; 2</math> = 2 <math>N</math> is not of the form <math>2^k - 1</math> when <math>NP = 1</math> = 3 <math>N</math> is not of the form <math>2^k</math> when <math>NP = 0</math> = 4 BLKTRI failed while computing results that depend on the coefficient arrays AN, BN, CN. Check these arrays and see the section on Special Conditions below. = 5 <math>IDIMY &lt; M</math></p>
	<p>W Contains intermediate values that must not be destroyed if BLKTRI will be called again with <math>IFLG = 1</math>.</p>
<u>Entry Points</u>	BLKTRI, BLKTR1, PROD, PRODP, CPROD, CPRODP, COMPB, IDX, PADD, PPADD
<u>Special Conditions</u>	The algorithm may fail if $ BM(I) + BN(J)  <  AM(I)  +  AN(J)  +  CM(I)  +  CN(J) $ for some I and J.
<u>Common Blocks</u>	CBLKT
<u>I/O</u>	None
<u>Precision</u>	Single
<u>Originator</u>	Paul N. Swarztrauber
<u>Space Required</u>	$3482_{10}$ locations on the NCAR Control Data 7600
<u>Accuracy and Timing</u>	The execution time is proportional to $MN \log_2 N$ and is tabulated below for the NCAR 7600 computer. To test the accuracy of the method a uniform random number generator

was used to create an array  $V(I,J)$ , where  $0 \leq V(I,J) \leq 1$ . An array  $F(I,J)$  was then computed by differencing  $V(I,J)$  in double precision using the difference equations that correspond to the sample problem at the end of this chapter. With  $F(I,J)$  as a right side, subroutine BLKTRI was used to compute a solution  $U(I,J)$ . The maximum absolute difference between  $U(I,J)$  and  $V(I,J)$  is tabulated below. These errors can be compared with the accuracy of the NCAR 7600 ( $10^{-14}$ ) to give a measure of the loss of significant digits due to roundoff. This error should not be confused with the discretization error which is given in the sample problem.

M	N	Execution Time (msec)	Relative Difference
31	31	54	$2.5 \times 10^{-13}$
63	63	264	$7.6 \times 10^{-13}$
127	127	1272	$5.7 \times 10^{-11}$

The execution times should be doubled if  $MP = 0$ .

Portability American National Standards Institute FORTRAN with no machine-dependent constants

Required Resident Routines SQRTF, CABS (complex absolute value)

#### APPLICATIONS TO DIFFERENCE APPROXIMATIONS

In this section we will describe how to use BLKTRI to solve separable elliptic equations. We will assume that the equation is defined on  $a < x < b$ ,  $c < y < d$  and has the form

$$a(x)\frac{\partial^2 u}{\partial x^2} + b(x)\frac{\partial u}{\partial x} + c(x)u + d(y)\frac{\partial^2 u}{\partial y^2} + e(y)\frac{\partial u}{\partial y} + f(y)u = g(x,y), \quad (1)$$

where  $a(x)d(y) > 0$ . Certain equations are not given in this form but can be put in this form, e.g., Poisson's equation in spherical coordinates. Assume

that some grid  $(x_i, y_j)$  is placed on the region, with equal grid spacings of  $\Delta x$  and  $\Delta y$ .

Define

$$\begin{aligned} a_i &= a(x_i) & d_j &= d(y_j) \\ b_i &= b(x_i) & e_j &= e(y_j) \\ c_i &= c(x_i) & f_j &= f(y_j) \end{aligned} \quad (2)$$

$$g_{i,j} = g(x_i, y_j) \quad (3)$$

$$u_{i,j} \approx u(x_i, y_j)$$

Then a finite difference approximation to Eq. (1) is

$$\begin{aligned} a_i \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{\Delta x^2} + b_i \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + c_i u_{i,j} \\ + d_j \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{\Delta y^2} + e_j \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \\ + f_j u_{i,j} = g_{i,j} \end{aligned} \quad (4)$$

Define

$$\begin{aligned} ax_i &= \frac{a_i}{\Delta x^2} - \frac{b_i}{2\Delta x} \\ bx_i &= \frac{-2a_i}{\Delta x^2} + c_i \\ cx_i &= \frac{a_i}{\Delta x^2} + \frac{b_i}{2\Delta x} \end{aligned} \quad (5)$$

$$\begin{aligned}ay_j &= \frac{d_j}{\Delta y^2} - \frac{e_j}{2\Delta y} \\by_j &= \frac{2d_j}{\Delta y^2} + f_j \\cy_j &= \frac{d_j}{\Delta y^2} + \frac{e_j}{2\Delta y}\end{aligned}\tag{6}$$

$$Pu_{i,j} = ax_{i-1,j}u_{i-1,j} + bx_iu_{i,j} + cx_{i+1,j}u_{i+1,j}\tag{7}$$

$$Qu_{i,j} = ay_{j-1}u_{i,j-1} + by_ju_{i,j} + cy_{j+1}u_{i,j+1}\tag{8}$$

Then Eq. (4) can be written

$$Pu_{i,j} + Qu_{i,j} = g_{i,j}\tag{9}$$

We will now discuss the incorporation of the boundary conditions. We will consider the x-direction only; however, the same analysis applies in the y-direction. There are four possible combinations: solution specified-solution specified, mixed boundary condition-solution specified, mixed boundary condition-mixed boundary condition, and periodic solution specified; the analysis at corners is also discussed.

#### 1. Solution specified-solution specified:

Here we require that  $u(a, y_j) = p_j$  and  $u(b, y_j) = q_j$ .

Define

$$\Delta x = \frac{b-a}{M+1}\tag{10}$$

$$x_i = i\Delta x + a \quad i = 0, 1, 2, \dots, M+1\tag{11}$$

At  $i = 1$  Eq. (4) has the form

$$ax_{1,0,j}u_{0,j} + bx_{1,1,j}u_{1,j} + cx_{1,2,j}u_{2,j} + Qu_{1,j} = g_{1,j}\tag{12}$$

The boundary condition specifies  $u_{0,j} = p_j$  and hence Eq. (12) can be written:

$$bx_1 u_{1,j} + cx_1 u_{2,j} + qu_{1,j} = g_{1,j} - ax_1 p_j . \quad (13)$$

Similarly, at  $i = M$  we obtain

$$ax_M u_{M-1,j} + bx_M u_{M,j} - g_{M,j} - cx_M q_j . \quad (14)$$

Using Eqs. (5), (13), and (14), we can determine the following quantities for input to BLKTRI.<sup>†</sup>

$$\begin{aligned} AM(1) &= 0 \\ BM(1) &= bx_1 \\ CM(1) &= cx_1 \\ Y(1,J) &= g_{1,j} - ax_1 p_j \quad J = j = 2, 3, \dots, N-1 \end{aligned} \quad (15)$$

$$\begin{aligned} AM(I) &= ax_i \\ BM(I) &= bx_i \\ CM(I) &= cx_i \quad I = i = 2, 3, \dots, M-1 \\ Y(I,J) &= g_{i,j} \quad J = j = 2, 3, \dots, N-1 \end{aligned} \quad (16)$$

$$\begin{aligned} AM(M) &= ax_M \\ BM(M) &= bx_M \\ CM(M) &= 0 \\ Y(M,J) &= g_{M,j} - cx_M q_j \quad J = j = 2, 3, \dots, N-1 \end{aligned} \quad (17)$$

See the discussion in section 5 below concerning the calculation of  $Y(I,J)$  at the corners.

## 2. Mixed boundary condition-solution specified:

Here we require  $\alpha u(a, y_j) + \frac{\partial}{\partial x} u(a, y_j) = p_j$ . The derivative boundary condition corresponds to the case  $\alpha = 0$ . At  $x = b$  we also require that  $u(b, y_j) = q_j$ .

---

<sup>†</sup> Capital letters such as AM, BM, CM, and Y refer to FORTRAN input arrays. Small letters such as  $ax_i$ ,  $bx_i$ , and  $cx_i$  refer to algebraic quantities.

Define

$$\Delta x = \frac{b-a}{M} \quad (18)$$

$$x_i = (i-1)\Delta x + a \quad i = 0, 1, 2, \dots, M+1 \quad . \quad (19)$$

Note that this definition of the grid differs from that given for condition 1 above. Note also that  $x_0 = a - \Delta x$  is outside  $[a, b]$ . This virtual point will be used as a computational device to center the boundary condition at  $x = a$  and thereby provide the same accuracy throughout the region  $R$ .

At  $i = 1$ , Eq. (4) has the form

$$ax_1 u_{0,j} + bx_1 u_{1,j} + cx_1 u_{2,j} + Qu_{1,j} = g_{1,j} \quad . \quad (20)$$

The discrete form of the boundary condition is

$$-\frac{1}{2\Delta x} u_{0,j} + \alpha u_{1,j} + \frac{1}{2\Delta x} u_{2,j} = p_j \quad . \quad (21)$$

Eliminating  $u_{0,j}$  between Eqs. (20) and (21), we obtain

$$\begin{aligned} & (bx_1 + ax_1 2\Delta x \alpha) u_{1,j} + (cx_1 + ax_1) u_{2,j} + Qu_{1,j} \\ & = g_{1,j} + ax_1 2\Delta x p_j \quad . \end{aligned} \quad (22)$$

At  $i = M$ , Eq. (4) has the form

$$ax_M u_{M-1,j} + bx_M u_{M,j} + cx_M u_{M+1,j} + Qu_{M,j} = g_{M,j} \quad . \quad (23)$$

The other boundary condition is

$$u_{M+1,j} = q_j \quad . \quad (24)$$

Hence Eq. (23) may be written

$$ax_M u_{M-1,j} + bx_M u_{M,j} + Qu_{M,j} = g_{M,j} - cx_M q_j \quad . \quad (25)$$

Using Eqs. (5), (22), and (25), we can determine the following quantities for input to BLKTRI.

$$\begin{aligned} AM(1) &= 0 \\ BM(1) &= bx_1 + ax_1^2 \Delta x \alpha \\ CM(1) &= cx_1 + ax_1 \\ Y(1,J) &= g_{1,j} + ax_1^2 \Delta x p_j \quad J = j = 2, 3, \dots, N-1 \end{aligned} \tag{26}$$

$$\begin{aligned} AM(I) &= ax_1 \\ BM(I) &= bx_i \quad I = i = 2, 3, \dots, M-1 \\ CM(I) &= cx_1 \\ Y(I,J) &= g_{i,j} \quad I = 2, \dots, M-1 ; J = 2, \dots, N-1 \end{aligned} \tag{27}$$

$$\begin{aligned} AM(M) &= ax_M \\ BM(M) &= bx_M \\ CM(M) &= 0 \\ Y(M,J) &= g_{M,j} - cx_M q_j \quad J = j = 2, 3, \dots, N-1 \end{aligned}$$

See the discussion in section 5 below concerning the calculation of  $Y(I,J)$  at the corners.

### 3. Mixed boundary condition-mixed boundary condition specified:

Here we require  $\alpha u(a, y_j) + \frac{\partial}{\partial x} u(a, y_j) = p_j$  and  $\beta u(b, y_j) + \frac{\partial}{\partial x} u(b, y_j) = q_j$ . The derivative boundary condition corresponds to the case  $\alpha = \beta = 0$ .

Define

$$\Delta x = \frac{b-a}{M-1} \tag{29}$$

$$x_i = (i-1)\Delta x + a \quad i = 0, 1, 2, \dots, M+1 \tag{30}$$

Note that this definition of the grid differs from that for conditions 1 and 2 above. Note too that points  $x_0$  and  $x_{M+1}$  lie a distance  $\Delta x$  outside the interval  $[a, b]$ . These virtual points will be used as a computational device to center the boundary conditions at  $x = a$  and  $x = b$ . This

maintains the same order of accuracy through the region R.

Following the analysis at  $i = 1$  in B, we observe that Eq. (22) also holds for the present case.

At  $i = M$ , Eq. (4) has the form

$$ax_M u_{M-1,j} + bx_M u_{M,j} + cx_M u_{M+1,j} + qu_M = g_{M,j} \quad . \quad (31)$$

The discrete form of the boundary condition is

$$-\frac{1}{2\Delta x} u_{M-1,j} + \beta u_{M,j} + \frac{1}{2\Delta x} u_{M+1,j} = q_j \quad . \quad (32)$$

Eliminating  $u_{M+1,j}$  between Eq. (31) and (32) we obtain

$$(ax_M + cx_M)u_{M-1,j} + (bx_M - cx_M^2\Delta x\beta)u_{M,j} + qu_M = g_{M,j} - cx_M^2\Delta xq_j \quad . \quad (33)$$

Using Eqs. (5), (22), and (33), we can determine the following quantities for input to BLKTRI

$$\begin{aligned} AM(1) &= 0 \\ BM(1) &= bx_1 + ax_1^2\Delta x\alpha \\ CM(1) &= cx_1 + ax_1 \\ Y(1,J) &= g_{1,j} + ax_1^2\Delta x p_j \quad J = j = 2,3,\dots,N-1 \end{aligned} \quad (34)$$

$$\begin{aligned} AM(I) &= ax_i \\ BM(I) &= bx_i \quad I = i = 2,3,\dots,M-1 \\ CM(I) &= cx_i \\ Y(I,J) &= g_{i,j} \quad I = 2,\dots,M-1 ; J = 2,\dots,N-1 \end{aligned} \quad (35)$$

$$\begin{aligned} AM(M) &= ax_M + cx_M \\ BM(M) &= bx_M - cx_M^2\Delta x\beta \\ CM(M) &= 0 \\ Y(M,J) &= g_{M,j} - cx_M^2\Delta x q_j \quad J = j = 2,3,\dots,N-1 \end{aligned} \quad (36)$$

See the discussion in section 5 below concerning the calculation of  $Y(I,J)$  at corners.

4. Periodic solution specified:

Here we require that  $u(a+x, y_j) = u(b+x, y_j)$  for all  $x$ .

Define

$$\Delta x = \frac{b-a}{M} \quad (37)$$

$$x_i = (i-1)\Delta x + a \quad i = 0, 1, 2, \dots, M+1 \quad . \quad (38)$$

At  $i = 1$ , Eq. (4) has the form of Eq. (20), but from the boundary condition we have

$$u_{0,j} = u_{M,j} \quad . \quad (39)$$

Therefore, Eq. (20) has the form

$$ax_1 u_{M,j} + bx_1 u_{1,j} + cx_1 u_{2,j} + qu_{1,j} = g_{1,j} \quad . \quad (40)$$

Similarly, at  $i = M$ , we use Eq. (31) and the boundary condition  $u_{M+1} = u_{1,j}$  to obtain

$$ax_M u_{M-1,j} + bx_M u_{M,j} + cx_M u_{1,j} + qu_{M,j} = g_{M,j} \quad . \quad (41)$$

Using Eqs. (5), (40), and (41), we can determine the following quantities for input to BLKTRI:

$$\begin{aligned} AM(I) &= ax_i \\ BM(I) &= bx_i \quad I = i = 1, 2, \dots, M \\ CM(I) &= cx_i \end{aligned} \quad . \quad (42)$$

Note that since  $AM(1)$  and  $CM(M)$  are not zero we must set  $MP = 0$  in the calling sequence.

$$Y(I,J) = g_{i,j} \quad I = 1, 2, \dots, M ; J = 2, \dots, N-1 \quad (43)$$

5. Treatment of  $Y(I,J)$  at corners:

For  $I = 2, \dots, M-1$  and  $J = 2, \dots, N-1$ , we set  $Y(I,J) = g_{i,j}$ . However, on the boundaries  $Y(I,J)$  will be modified as a result of the boundary conditions (except for the periodic case). For example, this

modification appears in the last of Eqs. (15). We note, however, that in Eqs. (15) the indices  $J = 1$  and  $J = N$  do not appear. This corresponds to a corner where the modification will include contributions from two boundary conditions. Consider the case of  $Y(1,1)$  where the boundary conditions  $u(a,y_j) = p_j$  and  $u(x_i,c) = q_i$

Then, Eq. (4) centered at  $i = 1, j = 1$  is

$$ax_1 u_{0,1} + bx_1 u_{1,1} + cx_1 u_{2,1} + ay_1 u_{1,0} + by_1 u_{1,1} + cy_1 u_{1,2} = g_{1,1} . \quad (44)$$

But  $u_{0,1} = p_1$  and  $u_{1,0} = q_1$ . Therefore, Eq. (44) has the form

$$bx_1 u_{1,1} + cx_1 u_{2,1} + by_1 u_{1,1} + cy_1 u_{1,2} = g_{1,1} - ax_1 p_1 - ay_1 q_1 . \quad (45)$$

Hence the input to BLKTRI is

$$Y(1,1) = g_{1,1} - ax_1 p_1 - ay_1 q_1 . \quad (46)$$

#### SAMPLE PROBLEM

In this section we will demonstrate how to use BLKTRI to solve the Poisson equation on a rectangle using "stretch coordinates." That is, the rectangle is transformed in order to resolve a particular portion such as a boundary layer.

Prior to the transformation, the problem is to determine an approximate solution to

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f(x,y) \quad a \leq x \leq b ; c \leq y \leq d$$

with the solution specified on the boundary.

If we let  $x = x(s)$  and  $y = y(t)$ , in terms of  $s$  and  $t$ , the problem is to determine an approximate solution to

$$\frac{1}{x} \frac{\partial}{\partial s} \left( \frac{1}{x} \frac{\partial u}{\partial s} \right) + \frac{1}{y} \frac{\partial}{\partial t} \left( \frac{1}{y} \frac{\partial u}{\partial t} \right) = f[x(s),y(t)] .$$

Assume that a boundary layer exists along  $s = 0$  and  $t = 0$ ; i.e., the solution varies rapidly in the neighborhood of the boundaries. We wish to select  $x(s)$  and  $y(t)$  so that in terms of  $s$  and  $t$  the solution is no longer rapidly varying, with the result that the finite difference approximation is accurate. The functions  $x(s)$  and  $y(t)$  are, therefore, best chosen when based on a priori knowledge of the behavior of the solution in the boundary layer. This is easily accomplished in the example that will be given, but difficult in practice. Possibly a more satisfactory approach in practice would be iterative. That is, select  $x(s)$ ,  $y(t)$  and then modify them based on the solution.

From several possible selections (see reference [3], we choose  $x(s) = s^2$ ,  $y(t) = t^2$ . The boundary data on the right side are specified so that  $u(x,y) = (xy)^{5/2} = s^5 t^5$  is a solution; i.e.,  $f(s,t) = \frac{15}{4}st(s^4+t^4)$ .

The boundary layer is exhibited by the fact that

$$\frac{\partial f}{\partial x}(0,y) = \frac{\partial f}{\partial y}(x,0) = \infty .$$

The transformed equation is

$$\frac{1}{2s} \frac{\partial}{\partial s} \left( \frac{1}{2s} \frac{\partial u}{\partial s} \right) + \frac{1}{2t} \frac{\partial}{\partial t} \left( \frac{1}{2t} \frac{\partial u}{\partial t} \right) = \frac{15}{4} st (s^4+t^4)$$

which we will assume holds on the rectangle  $0 < s < 1$  and  $0 < t < 1$ .

We further specify the boundary data

$$u(0,t) = 0 \quad u(1,t) = t^5, \quad 0 \leq t \leq 1 ,$$
$$u(s,0) = 0 \quad u(s,1) = s^5, \quad 0 \leq s \leq 1 .$$

To solve the problem numerically we first select integers  $M$  and  $N$  which define grid spacings  $\Delta s = 1/(M+1)$   $\Delta t = 1/(N+1)$  and the grid

$$s_i = i\Delta s \quad i = 0, \frac{1}{2}, 1, \dots, M+1$$

$$t_j = j\Delta t \quad j = 0, \frac{1}{2}, 1, \dots, N+1 .$$

In order to determine an approximate solution  $u_{i,j} \approx u(s_i, t_j)$  to the transformed equation we require  $u_{i,j}$  to satisfy the following finite difference approximation

$$\frac{1}{2s_i \Delta s} \left[ \frac{(u_{i+1,j} - u_{i,j})}{2s_{i+\frac{1}{2}} \Delta s} - \frac{(u_{i,j} - u_{i-1,j})}{2s_{i-\frac{1}{2}} \Delta s} \right] + \frac{1}{2t_j \Delta t} \left[ \frac{(u_{i,j+1} - u_{i,j})}{2t_{j+\frac{1}{2}} \Delta t} - \frac{(u_{i,j} - u_{i,j-1})}{2t_{j-\frac{1}{2}} \Delta t} \right] = \frac{15}{4} s_i t_j (s_i^4 + t_j^4)$$

subject to the boundary data

$$u_{0,j} = 0 ; \quad u_{M+1,j} = t_j^5 \quad j = 0, \dots, N+1$$

$$u_{i,0} = 0 ; \quad u_{i,N+1} = s_i^5 \quad k = 0, \dots, M+1 .$$

The following input to subroutine BLKTRI was obtained by using the methods described in the previous section

For  $I = i = 1, 2, \dots, M$

$$AM(I) = \frac{1}{2s_i \Delta s} \cdot \frac{1}{2s_{i-\frac{1}{2}} \Delta s}$$

$$BM(I) = \frac{-1}{2s_i \Delta s} \left( \frac{1}{2s_{i+\frac{1}{2}} \Delta s} + \frac{1}{2s_{i-\frac{1}{2}} \Delta s} \right)$$

$$CM(I) = \frac{1}{2s_i \Delta s} \cdot \frac{1}{2s_{i+\frac{1}{2}} \Delta s}$$

except for  $AM(1) = CM(M) = 0$ .

For  $J = j = 1, 2, \dots, N$

$$AN(J) = \frac{1}{2t_j \Delta t} \cdot \frac{1}{2t_{j-\frac{1}{2}} \Delta t}$$

$$BN(J) = \frac{-1}{2t_j \Delta t} \left( \frac{1}{2t_{j+\frac{1}{2}} \Delta t} + \frac{1}{2t_{j-\frac{1}{2}} \Delta t} \right)$$

$$CN(J) = \frac{1}{2t_j \Delta t} \cdot \frac{1}{2t_{j+\frac{1}{2}} \Delta t}$$

except for  $AN(1) = CN(N) = 0$ .

For  $I = i = 1, \dots, M-1$  and  $J = j = 1, \dots, N-1$

$$Y(I,J) = \frac{15}{4} s_i t_j (s_i^4 + t_j^4) .$$

However, next to the boundaries  $s = 1$  and  $t = 1$ ,  $Y$  must be adjusted since the boundary data is non zero.

For  $J = j = 1, \dots, N-1$

$$Y(M,J) = \frac{15}{4} t_j (1+t_j^4) - \frac{1}{2s_M \Delta s} \cdot \frac{1}{2s_{M+\frac{1}{2}} \Delta s} t_j^5$$

and for  $I = i = 1, \dots, M-1$

$$Y(I,N) = \frac{15}{4} s_i (s_i^4 + 1) - \frac{1}{2t_N \Delta t} \cdot \frac{1}{2t_{N+\frac{1}{2}} \Delta t} s_i^5 .$$

Finally at the corner  $I = N$   $J = M$

$$Y(M,N) = \frac{15}{2} - \frac{1}{2t_N \Delta t} \cdot \frac{1}{2t_{N+\frac{1}{2}} \Delta t} s_M^5 - \frac{1}{2s_M \Delta s} \cdot \frac{1}{2s_{M+\frac{1}{2}} \Delta s} t_N^5 .$$

The other parameters are IFLG = 0 for initialization with a subsequent call with IFLG = 1 to obtain the solution. Both MP and NP are 1.

The following FORTRAN program illustrates in detail the solution of the problem.

```
C
C      PROGRAM TO ILLUSTRATE THE USE OF BLKTRI
C
C      DIMENSION      Y(75,105)      ,AM(75),BM(75),CM(75),AN(105)      ,
C      1              BN(105)      ,CN(105)      ,W(952),S(75)  ,T(105)
C
C      THE VALUE OF IDIMY IS THE FIRST DIMENSION OF Y. SINCE NP=1 W IS
C      DIMENSIONED 2*(N+2)(LOG2(N+1)-1)+6M+2)=952
C
C      IFLG = 0
C      NP = 1
C      N = 63
C      MP = 1
C      M = 50
C      IDIMY = 75
C
C      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING THE
C      COEFFICIENTS AND THE ARRAY Y.
```

```
C
      DELTAS = 1./(M+1)
      DO 100 I=1,M
          S(I) = I*DELTAS
100 CONTINUE
      DELTAT = 1./(N+1)
      DO 105 J=1,N
          T(J) = J*DELTAT
105 CONTINUE
C
C      COMPUTE THE COEFFICIENTS AM,BM,CM CORRESPONDING TO THE S DIRECTION
C
      HDS = DELTAS/2.
      TDS = DELTAS+DELTAS
      DO 110 I=1,M
          TEMP1 = 1./(S(I)*TDS)
          TEMP2 = 1./*((S(I)-HDS)*TDS)
          TEMP3 = 1./*((S(I)+HDS)*TDS)
          AM(I) = TEMP1*TEMP2
          CM(I) = TEMP1*TEMP3
          BM(I) = -(AM(I)+CM(I))
110 CONTINUE
C
C      COMPUTE THE COEFFICIENTS AN,BN,CN CORRESPONDING TO THE T DIRECTION
C
      HDT = DELTAT/2.
      TDT = DELTAT+DELTAT
      DO 115 J=1,N
          TEMP1 = 1./*(T(J)*TDT)
          TEMP2 = 1./*((T(J)-HDT)*TDT)
          TEMP3 = 1./*((T(J)+HDT)*TDT)
          AN(J) = TEMP1*TEMP2
          CN(J) = TEMP1*TEMP3
          BN(J) = -(AN(J)+CN(J))
115 CONTINUE
C
C      COMPUTE RIGHT SIDE OF EQUATION
C
      DO 125 J=1,N
          DO 120 I=1,M
              Y(I,J) = 3.75*S(I)*T(J)*(S(I)**4 + T(J)**4)
120      CONTINUE
125 CONTINUE
C
C      INCLUDE NONHOMOGENEOUS BOUNDARY INTO RIGHT SIDE. NOTE THAT THE
C      CORNER AT J=N, I=M INCLUDES CONTRIBUTIONS FROM BOTH BOUNDARIES.
C
      DO 130 J=1,N
          Y(M,J) = Y(M,J) - CM(M)*T(J)**5
130 CONTINUE
```

```
DO 135 I=1,M
    Y(I,N) = Y(I,N) - CN(N)*S(I)**5
135 CONTINUE
C
140 CALL BLKTRI (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDLIMY,Y,IERROR,W)
    IFLG = IFLG+1
    IF (IFLG-1) 140,140,145
C
C      COMPUTE DISCRETIZATION ERROR
C
145 ERR = 0.
    DO 155 J=1,N
        DO 150 I=1,M
            Z = ABS(Y(I,J)-(S(I)*T(J))**5)
            IF (Z .GT. ERR) ERR = Z
150     CONTINUE
155     CONTINUE
    PRINT 1001 , IERROR,ERR
    STOP
C
1001 FORMAT (//9H IERROR = I3,10X,22HDISCRETIZATION ERROR =E12.4)
C
END
```

After the CALL to BLKTRI, the discretization error--the maximum absolute difference between the exact solution and the finite difference approximation--is computed and printed. The output line is:

IERROR = 0                    DISCRETIZATION ERROR = 1.6478E-05



## APPENDIX

### THE LEAST SQUARES SOLUTION OF SINGULAR LINEAR SYSTEMS OF EQUATIONS

Certain problems described in this report do not have solutions in the usual sense; rather solutions are obtained in the sense of least squares. The purpose of this appendix is to describe this procedure in some detail.

The difficulty can be illustrated by the following continuous problem. Assume that a solution  $u(\theta, \phi)$  exists to Eq. (1) of Chapter V with  $\lambda = 0$ . If we multiply that equation by  $\sin\theta$  and integrate over the entire surface of the sphere we find that the left side is zero with the result

$$0 = \int_0^{2\pi} \int_0^{\pi} \sin\theta f(\theta, \phi) d\theta d\phi \quad . \quad (1)$$

Hence, if we assume that a solution exists,  $f(\theta, \phi)$  must satisfy Eq. (1). Equivalently, if  $f(\theta, \phi)$  does not satisfy Eq. (1), a solution does not exist.

For the discrete problem, there are analogous results. The problem consists of determining a solution to a linear system  $Au = f$  where  $A$  is a large sparse  $n \times n$  matrix that contains the coefficients of the finite difference equations. The components of the vector  $u$  are the approximate values of the solution at the grid points and the components of  $f$  are the values of the right side of the difference equation at the grid points. The linear system also includes the equations which result from the discretization of the boundary conditions. We now give a fundamental theorem of linear algebra, which will be proven later.

#### Theorem 1

A linear system of equations  $Au = f$  has a solution if and only if  $h^T f = 0$  for all  $h$  such that  $A^T h = 0$ .

Corresponding to each singular system discussed in this note there is only one independent vector  $h$  such that  $A^T h = 0$ , and it has the following

properties:

- P<sub>1</sub> If each row of A is multiplied by the corresponding component of h, the resulting matrix is symmetric; i.e., define a diagonal matrix  $H = \text{diag}(h^T)$ , then  $HA = (HA)^T = A^T H$ .
- P<sub>2</sub> All components of h are greater than zero, with the result that H is positive definite.
- P<sub>3</sub> Except for scalar multiples, h is the only vector such that  $A^T h = 0$ .

The vector h is given for each case in the chapter sections entitled Singular Systems.

From P<sub>3</sub> and Theorem 1 we determine that the system  $Au = f$  has a solution if and only if

$$h^T f = 0 \quad (2)$$

which is analogous to the restriction, Eq. (1) above, in the continuous case.

In practice, as a result of computational or observational errors, f will not satisfy Eq. (2) and hence a solution to  $Au = f$  will not exist. In this event, a reasonable alternative is to determine a least squares solution. However, before proceeding, we require three definitions:

D<sub>1</sub> Define the vector  $e^T = (1, 1, \dots, 1)$  of length n, whose components are all 1.

D<sub>2</sub> Define the inner product  $(s, t)_H$  of two arbitrary vectors s and t by

$$(s, t)_H = s^T H t \quad (3)$$

with induced norm

$$\|t\|_H^2 = (t, t)_H \quad .$$

D<sub>3</sub> Define the vector g as

$$g = f - \frac{h^T f}{h^T e} e \quad . \quad (4)$$

Note that since  $h^T g = 0$ , the system  $Au = g$  has a solution. The parameter PERTRB which is returned by the subroutines contains the value  $h^T f / h^T e$ .

We now state Theorem 2, which will be proven later.

Theorem 2

If  $u$  is a solution to the system  $Au = g$  where  $g$  is defined in Eq. (4) above, then  $u$  is a weighted least squares solution to the system  $Au = f$ ; i.e.,  $u$  minimizes  $\|Au - f\|_H$ .

The procedure therefore consists of first determining  $g$  from Eq. (4) and then determining  $u$  by solving the system  $Au = g$ . Additional details for a specific problem are contained in references [6,8].

We will now prove the theorems stated above.

Proof of Theorem 1:

Assume first that a solution  $u$  exists. Then

$$h^T Au = h^T f ,$$

but since  $h^T A = (A^T h)^T = 0$  ,

therefore  $h^T f = 0$  .

Conversely, assume that  $h^T f = 0$  for all  $h$  such that  $A^T h = 0$ . Let  $N(A^T)$  denote the space of all such vectors  $h$  and let  $k$  denote its dimension. Let  $n$  denote the order of  $A$ . Then the rank of  $A$  is  $n - k$ .

Select a basis for  $N(A^T)$  and row reduce it to echelon form, i.e., obtain a basis  $w_1, w_2, \dots, w_k$  of the form

$$w_\ell^T = (0, \dots, 0, 1, w_{\ell, i_\ell+1}, w_{\ell, i_\ell+2}, \dots, w_{\ell, n}) \quad \ell = 1, \dots, k \quad (5)$$

where the 1 is the  $i_\ell^{\text{th}}$  component and  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Then denote the remaining integers by  $j_m$   $m = 1, \dots, n-k$ . Therefore, the set of integers 1 through  $n$  is just the union of the disjoint sets

$\{i_\ell, \ell=1, \dots, k\}$  and  $\{j_m, m=1, \dots, n-k\}$ .

length  $n$

$$e_m^T = (0, \dots, 0, 1, 0, \dots, 0) \quad m = 1, \dots, n-k$$

where the 1 is the  $j_m^{\text{th}}$  component. Next form the  $n \times n$  matrix

$$R = \begin{bmatrix} w_1^T \\ \vdots \\ w_k^T \\ e_1^T \\ \vdots \\ \vdots \\ e_{n-k}^T \end{bmatrix} \quad (6)$$

By a simple permutation of rows, an upper triangular matrix can be produced with ones on the diagonal. Hence the determinant of  $R$  is  $\pm 1$ . and  $R$  is nonsingular. Therefore, the system  $Au = f$  has a solution if and only if  $RAu = Rf$  has a solution. Hence, we consider now the solution of the latter system. Since  $RA = (A^T R^T)^T$ , the first  $k$  rows of  $RA$  are zero. Also the first  $k$  components of  $Rf$  are zero by assumption. If  $a_i$  denotes the  $i$ th row of  $A$  and  $f_i$  denotes the  $i$ th component of  $f$ , then  $RAu = Rf$  has the form

$$\begin{bmatrix} 0 \\ \vdots \\ 0 \\ a_{j_1} \\ \vdots \\ \vdots \\ a_{j_{n-k}} \end{bmatrix} u = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_{j_1} \\ \vdots \\ \vdots \\ f_{j_{n-k}} \end{bmatrix} \quad (7)$$

Since the rank of  $A$  is  $n-k$  and  $R$  is nonsingular the rank of  $RA$  is  $n-k$  and therefore the last  $n-k$  rows of  $RA$  are independent. The  $(n-k) \times n$  rectangular subsystem can therefore be solved for  $u$ . For  $k > 0$  the

system is underdetermined and hence the solution is not unique. The steps of the proof may be retraced to determine that  $u$  is a solution of  $Au = f$ .

Proof of Theorem 2:

$$A^T H A u = A^T H g$$

$$= A^T H \left( f - \frac{h^T f}{h^T e} e \right) ,$$

$$\text{but } A^T H e = A^T h = 0 ,$$

$$\text{hence } A^T H A u = A^T H f . \quad (8)$$

However,  $u$  minimizes  $\|Au - f\|_H$  if and only if  $u$  is a solution to the normal Eq. (8).



REFERENCES

- [1] Buzbee, B.L., F.W. Dorr, J.A. George, and G.H. Golub, 1971: The direct solution of the discrete Poisson equation on irregular regions. *SIAM J. Numer. Anal.* 8, 722-736.
- [2] Buzbee, B.L., G.H. Golub, and C.W. Nielson, 1971: On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.* 7, 627-656.
- [3] de Rivas, E.K., 1972: On the use of non-uniform grids in finite difference equations. *J. Comput. Phys.* 10, 202-210.
- [4] Dorr, F.W., 1970: The direct solution of the discrete Poisson equation on a rectangle. *SIAM Review* 12, 248-263.
- [5] Hockney, R.W., 1970: The potential calculation and some applications. *Meth. in Comput. Phys.* 9, 135-211.
- [6] Swarztrauber, P.N., 1974: The direct solution of the discrete Poisson equation on the surface of a sphere. *J. Comput. Phys.* 15, 46-54.
- [7] \_\_\_\_\_: A direct method for the discrete solution of separable elliptic equations. *SIAM J. Numer. Anal.* 11, 1136-1150.
- [8] \_\_\_\_\_, and R.A. Sweet, 1973: The direct solution of the discrete Poisson equation on a disk. *SIAM J. Numer. Anal.* 10, 900-907.
- [9] Sweet, R.A., 1973: Direct methods for the solution of Poisson's equation on a staggered grid. *J. Comput. Phys.* 12, 422-428.
- [10] \_\_\_\_\_, 1974: A generalized cyclic reduction algorithm. *SIAM J. Numer. Anal.* 11, 506-520.



Version 2, October 1976

UPDATE AND ERRATA

Efficient FORTRAN Subprograms for the Solution  
of Elliptic Partial Differential Equations

by

Paul Swarztrauber and Roland Sweet

Technical Note TN/IA-109 July 1975  
National Center for Atmospheric Research  
Boulder, Colorado 80307

1. p. v, pgh. 2, line 3, should read:  
The first five subroutines solve a modified Helmholtz...
2. p. v, pgh. 3, line 3, should read:  
transforming in the third variable and using the parameter in the modified...
3. p. vi, pgh. 1, lines 3 and 4:  
replace "Helmholtz" by differential
4. p. vii, line 1:  
delete the phrase "on punched cards"
5. p. 4, 21, 41, 64, and 85, following the line: F must be dimensioned at least  $(M+1) \times (N+1)$ , insert:  
NOTE: If the table calls for both the solution u and right side f at the corner, then the solution must be specified.
6. p. 5, 23, 43:  
add to the list of entry points the names: POIS, POISGN, POINIT, TRIDP, TRID.
7. p. 5, 22, 42, 65, and 85:  
In the description of the parameter PERTRB, replace the last sentence with the following:  
This solution plus any constant is also a solution; hence, the solution is not unique. The value of PERTRB should be small compared to the right side F. Otherwise, a solution is obtained to an essentially different problem. This comparison should always be made to insure that a meaningful solution has been obtained.



8. p. 5, 23, 43, 66, 86, 98, 117:  
change Entry Points to Subprograms Required.
9. p. 6, line 6, should read:  
 $3677_{10}$  locations...
10. p. 22, line 6:  
"periodic" is misspelled
11. p. 23, line 10, should read:  
 $3974_{10}$  locations
12. p. 37, 60, first sentence should read:  
modified Helmholtz
13. p. 37, 60, just above the PARAMETER DESCRIPTION and PROGRAM SPECIFICATIONS, insert:  
This two-dimensional modified Helmholtz equation results from the Fourier transform of the three-dimensional Poisson equation.
14. p. 43, ninth line from the bottom should read:  
 $3794_{10}$  locations
15. p. 58, fourth line after statement number 145 should read:  
 $X = ABS(F(I,J)-(R(I)*Z(J))^{**4})$
16. p. 60, line 18 should read:  
A call with INTL=0 takes approximately 1.5 times...
17. p. 64, last line should read:  
 $2(L+1)(K-1) + 6(M+N) + MAX(2N,6M) + 14.$
18. p. 65, line 18 should read:  
 $= 1 \quad TS < 0 \text{ or } TF > \pi$
19. p. 65, after 8th line from the bottom, insert:  
 $-12 \quad TS = 0 \text{ and } MBDCND = 3, 4 \text{ or } 8 \quad \text{or}$   
 $TF = \pi \text{ and } MBDCND = 2, 3 \text{ or } 6$
20. p. 66:  
add to the list of entry points: BLKTRI, BLKTRII, PROD, PRODP,  
CPROD, CPRODP, COMPB, PPADD, PSGF, BSRH, PPSGF, PPSPF, TQLRAT,  
STORE, APXEPS.



21. p. 66, Common Blocks should read:

Common Blocks CBLKT, VALUE

22. p. 66, line 8, should read:

4907<sub>10</sub> locations...

23. p. 66, Accuracy and Timing section, line 2, following the first sentence, insert the sentence:

The times in the table do not include the time for initialization.

24. p. 77, last two lines, should read:

C THEREFORE K=5 AND W IS DIMENSIONED 2(L+1)(K-1)+6(M+N)+MAX(2N,6M)+14  
C = 902 (M=36)

25. p. 81, change the description of the parameter INTL to read:

= 0 on initial entry to PWSSSP or if PS, PF, N or NBDCND are changed from the previous call

= 1 if PS, PF, N and NBDCND are unchanged from a previous call

26. p. 85, third line from bottom should read:

= 1 TS < 0 or TF > π

27. p. 86, after seventh line insert:

= 10 TS = 0 and MBDCND = 3, 4 or 8 or  
TF = π and MBDCND = 2, 3 or 6

28. p. 86:

add to the list of entry points the names: POIS, POISGN, POINIT, TRIDP,  
TRID, APXEPS, STORE, NCHECK

29. p. 86, Common Blocks should read:

Common Blocks VALUE

30. p. 86, tenth line from the bottom should read:

4416<sub>10</sub> locations...

31. p. 87:

put M over the first column in the table

32. p. 88, sixth line from the bottom:

both "u<sub>i,j</sub>" should be replaced by u<sub>1,j</sub>

33. p. 94:

insert a comma before SINP(73) in the DIMENSION statement



34. p. 97, line 8:  
replace "A(M)" by C(M)
35. p. 97, line 9, should read:  
 $\neq 0$  if  $A(1) = C(M) = 0$
36. p. 98, line 15, should read:  
 $3033_{10}$  locations...
37. p. 112, line 10:  
add ,C(20) after "Y(120)"
38. p. 115, sixth line from the bottom, should read:  
NOTE: A call with IFLG=0 takes about half as much time...
39. p. 116, line 17, should read:  
...any integer greater than 4.
40. p. 116, third line from the bottom, should read:  
...+ MAX(4N,6M)].
41. p. last line, should read:  
 $[2(N+1)(\log_2(N+1) - 1) + \text{MAX}(2N,6M) + 2]$
42. p. 117, line 7, should read:  
 $= 1 \quad M < 5$
43. p. 117, the list of entry points should read:  
BLKTRI, BLKTRII, PROD, PRODP, CPROD, CPRODP, COMPB, PPADD, PSGF,  
BSRH, PPSGF, PPSPF, TQLRAT, STORE, APXEPS
44. p. 117, add to the list of Common Blocks:  
,VALUE
45. p. 117, fourth line from bottom, should read:  
 $3554_{10}$  locations...
46. p. 127, line 11  
a right parenthesis should follow the [3]
47. p. 129, eleventh line from the bottom, should read:  
C DIMENSIONED  $2*(N+1)(\log_2(N+1) - 1) + 2 + \text{MAX}(2N,6M) = 942$
48. p. 129, fourteenth line from the bottom, should read:  
1 BN(105) ,CN(105) ,W(942),S(75),T(105)



49. p. 130, eleventh line from the bottom, should read:

DO 120 I=1,M

50. p. 135, fifth line from bottom:

the " $w_{i,n}$ " at the end of the parenthesis should be  $w_{\ell,n}$

