Fishpack Documentation Notes

The most complete source of documentation for FISHPACK90 is this present document. If prospective users understand that the interfaces are different, they can learn about FISHPACK90 from earlier, more copious FISHPACK documentation. Here are the <u>details of the changes between FISHPACK and FISHPACK90</u>.

This <u>older FISHPACK document</u> contains theoretical discussion not available elsewhere. However, readers need to adapt the information to FISHPACK90 by understanding the following:

- name changes e.g. PWSCRT instead of HWSCRT;
- only 7 of the 19 solvers are discussed; and
- FISHPACK90 routines have a different interface.

Obtaining Software and Documentation

Programs, solvers and support files including some documentation are available at the download tab on the top of the NCAR FISHPACK90 home page. This distribution is slanted towards users running Linux, Mac, or Unix systems with a Fortran compiler, since the required utilities uncompress, tar, make etc are more likely available on those systems.

Solver Overview

The following table summarizes the contents of FISHPACK90. Descriptions can be obtained by clicking on the solver name. De cri PDEs solved are included after the table.

An Overview of FISHPACK90 Solvers

computation	subprogram	test program
2D Helmholtz in Cartesian coordinates (centered grid)	<u>hwscrt</u>	thwscrt
2D Helmholtz in polar coordinates (centered grid)	<u>hwsplr</u>	thwsplr
2D Helmholtz in cylindrical coordinates (centered grid)	hwscyl	thwscyl
2D Helmholtz in spherical coordinates (centered grid)	<u>hwsssp</u>	<u>thwsssp</u>
2D Helmholtz in spherical coordinates (centered grid, axisymmetric)	hwscsp	<u>thwscsp</u>
2D Helmholtz in Cartesian coordinates (staggered grid)	<u>hstcrt</u>	thstcrt
2D Helmholtz in polar coordinates (staggered grid)	<u>hstplr</u>	thstplr
2D Helmholtz in cylindrical coordinates (staggered grid)	<u>hstcyl</u>	thstcyl
2D Helmholtz in spherical coordinates (staggered grid)	<u>hstssp</u>	thstssp
2D Helmholtz in spherical coordinates (staggered grid, axisymmetric)	<u>hstcsp</u>	thstcsp
3D Helmholtz in Cartesian coordinates (centered grid)	hw3crt	thw3crt
2D General Separable PDE (second or fourth order, centered grid)	<u>sepeli</u>	tsepeli
2D Separable PDE (second or fourth order, centered grid)	sepx4	tsepx4
real linear systems solver (centered grid, sepx4)	<u>genbun</u>	<u>tgenbun</u>
real block tridiagonal linear systems solver (centered grid, sepeli)	<u>blktri</u>	<u>tblktri</u>
real linear systems solver (staggered grid)	poistg	tpoistg
real linear systems solver (3D, centered grid)	pois3d	tpois3d

complex linear systems solver (centered grid)	<u>cmbnbn</u>	tcmgnbn
complex block tridiagonal linear systems solver (centered grid)	<u>cblktri</u>	tcblktri
real and complex fft package	fftpack	use with 3D solvers

DESCRIPTION

The form of the elliptic equations approximated are outlined below. The solvers allow periodic, specified, or derivative boundary conditions.

hwscrt

Subroutine for solving the standard five-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a centered finite difference grid.

(d/dx)(du/dx) + (d/dy)(du/dy) + lambda*u = f(x,y)

Additional files required: genbun, gnbnaux, comf

Sample program file: thwscrt

<u>hwsplr</u>

Subroutine for solving a five-point finite difference approximation to the Helmholtz equation in polar coordinates using a centered finite difference grid.

(1/r)(d/dr)(r(du/dr)) + (1/r**2)(d/dtheta)(du/dtheta) + lambda*u = f(r,theta)

Additional files required: genbun, gnbnaux, comf

Sample program file: thwsplr

hwscyl

Subroutine for solving a five-point finite difference approximation to the modified Helmholtz equation in cylindrical coordinates using a centered finite difference grid.

```
(1/r)(d/dr)(r(du/dr)) + (d/dz)(du/dz) + (lambda/r**2)*u = f(r,z)
```

Additional files required: genbun, gnbnaux, comf

Sample program file: thwscyl

hwsssp

Subroutine for solving a five-point finite difference approximation to the Helmholtz equation in spherical coordinates and on the surface of the unit sphere using a centered finite difference grid

```
(1/sin(theta))(d/dtheta)(sin(theta)(du/dtheta)) + (1/sin(theta)**2)(d/dphi)(du/dphi) + lambda*u = f(theta,phi)
```

Additional files required: genbun, gnbnaux, comf

Sample program file: thwsssp

Subroutine for solving a five-point finite difference approximation to the modified Helmholtz equation in spherical coordinates assuming axisymmetry (no dependence on longitude) using a centered finite difference grid.

```
(1/r^{**}2)(d/dr)(r^{**}2(du/dr)) +

1/(r^{**}2^{*}sin(theta))(d/dtheta)(sin(theta)(du/dtheta)) +

(lambda/(r^{*}sin(theta)^{**}2))^{*}u = f(theta,r)
```

Additional files required: **blktri**, **comf**

Sample program file: thwscsp

hstcrt

Subroutine for solving the standard five-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a staggered finite difference grid

```
(d/dx)(du/dx) + (d/dy)(du/dy) + lambda*u = f(x,y)
```

Additional files required: genbun, poistg, gnbnaux, comf

Sample program file: thstcrt

<u>hstplr</u>

Subroutine for solving a five-point finite difference approximation to the Helmholtz equation in polar coordinates using a staggered finite difference grid

```
(1/r)(d/dr)(r(du/dr)) + (1/r^{**}2)(d/dtheta)(du/dtheta) + lambda^*u = f(r,theta)
```

Additional files required: genbun, poistg, gnbnaux, comf

Sample program file: thstplr

hstcyl

Subroutine for solving a five-point finite difference approximation to the modified Helmholtz equation in cylindrical coordinates using a staggered finite difference grid.

```
(1/r)(d/dr)(r(du/dr)) + (d/dz)(du/dz) + (lambda/r**2)*u = f(r,z)
```

Additional files required: genbun, poistg, gnbnaux, comf

Sample program file: thstcyl

<u>hwsssp</u>

Subroutine for solving a five-point finite difference approximation to the Helmholtz equation in spherical coordinates and on the surface of the unit sphere using a staggered finite difference grid

```
(1/sin(theta))(d/dtheta)(sin(theta)(du/dtheta)) + (1/sin(theta)**2)(d/dphi)(du/dphi) + lambda*u = f(theta,phi)
```

Additional files required: genbun, poistg, gnbnaux, comf

Sample program file: thstssp

<u>hstcsp</u>

Subroutine for solving a five-point finite difference approximation to the modified Helmholtz equation in spherical coordinates assuming axisymmetry (no dependence on longitude) using a staggered finite difference grid.

```
(1/r^*2)(d/dr)(r^*2(du/dr)) +

1/(r^*2*sin(theta))(d/dtheta)(sin(theta)(du/dtheta)) +

(lambda/(r^*sin(theta)^*2))^*u = f(theta,r)
```

Additional files required: **blktri**, **comf**

Sample program file: thwscsp

hw3crt

Subroutine for solving the standard seven-point finite difference approximation to the Helmholtz equation in cartesian coordinates using a centered finite difference grid.

```
(d/dx)(du/dx) + (d/dy)(du/dy) + (d/dz)(du/dz) + lambda^*u = f(x,y,z)
```

Additional files required: pois3d, comf, fftpack

Sample program file: thw3crt

Subroutine for automatically discretizing and solving second and (optionally) fourth order finite difference approximations on a uniform grid to certain separable elliptic partial differential equations with constant coefficients in one direction on a rectangle.

$$a(x)(d/dx)(du/dx) + b(x)du/dx + c(x)u + (d/dy)(du/dy) = g(x,y)$$

Additional files required: genbun, gnbnaux, comf

Sample program file: tsepx4

<u>sepeli</u>

Subroutine for automatically discretizing and solving second and (optionally) fourth order finite difference approximations on a uniform grid to the general separable elliptic partial differential equation on a rectangle.

$$a(x)(d/dx)(du/dx) + b(x)du/dx + c(x)u + d(y)(d/dy)(du/dy) + e(y)du/dy + f(y)u = g(x,y)$$

Additional files required: blktri, comf

Sample program file: tsepeli

genbun

Subroutine for solving the real linear system of equations that results from a finite difference approximation on a centered grid to certain two-dimensional elliptic partial differential equations (e.g., see **sepx4**) with constant coefficients in one direction.

Additional files required: gnbnaux, comf

Sample program file: tgenbun

blktri

Subroutine for solving block tridiagonal linear systems that arise from finite difference approximations to separable two-dimensional elliptic partial differential equations (see **sepeli**).

Additional files required: **comf** Sample program file: <u>tblktri</u>

poistg

Subroutine for solving a block tridiagonal linear system of equations that arises from finite difference approximations on a staggered grid to two-dimensional elliptic partial differential equations with constant coefficients in one direction.

Additional files required: **gnbnaux**, **comf**

Sample program file: tpoistg

Subroutine for solving a block tridiagonal linear system of equations that arises from finite difference approximations to three-dimensional elliptic partial differential equations in a box.

Additional files required: **comf**, **fftpack**

Sample program file: tpois3d

<u>cmgnbn</u>

Subroutine for solving a complex block tridiagonal linear system arising from finite difference approximations to separable complex two-dimensional elliptic partial differential equations. box.

Additional files required: **comf** Sample program file: <u>tcmgnbn</u>

cblktri

Subroutine for solving a complex block tridiagonal linear system of equations arising from finite difference approximation to separable complex two-dimensional elliptic partial differential equations.

Additional files required: **comf** Sample program file: <u>tcblktri</u>

Details of the changes between FISHPACK and FISHPACK90

Details of FISHPACK90 changes

SCD released version 1.1 of FISHPACK90 on December 19, 2005. It is an improvement over version 1.0 in that it uses Fortran90 intrinsic function ASSOCIATED to check on pointer association. Also, version 1.1 offers an integated Makefile for use with gmake.

Here is some history for version 1.0:

SCD released FISHPACK90 1.0 on September 30 2004. It is an improvement of the original Fortran77 FISHPACK insofar as it has removed workspace arguments in the solvers, replacing them with FORTRAN90 derived data types that are pointers to real and complex allocatable arrays and are opaque to user level interface. And also this version replaced many internal interfaces in conformance with strict prototype matching of Fortran90.

Neither version FISHPACK90 1.0 nor 1.1 are full-blown Fortran90 implementation of FISHPACK. User calls to the old FISHPACK solvers are not compatible with calls to FISHPACK90 solvers.

IMPORTANT NOTE: FISHPACK90 has dependencies in the FFTPACK library; the present version of FFTPACK has not been updated in the same manner as FISHPACK90. That is, FFTPACK may not strictly conform to the Fortran90 interface specification.

The workspace changes in FISHPACK90 eliminate a mixed-mode conflict that occurred in the original FISHPACK. These changes simplify the user interface required to call the solvers. Details of these changes are described below.

All of the files in the original FISHPACK retain the same names in FISHPACK90. However, all of the FISHPACK90 solvers require loading the new FORTRAN 90 module file "fish.f." The 14 solvers

```
cmgnbn.f,genbun.f,hstcrt.f,hstcyl.f,hst
plr.f,hstssp.f,hw3crt.f,
```

```
hwscrt.f,hwscyl.f,hwsplr.f,hwsssp.f,poisd.f,poistg.f,sepx4.f
```

have the same arguments in both packages, *except* that the workspace argument has been deleted in FISHPACK90. For example, in the original FISHPACK:

```
CALL HSTCRT

(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,

+
ELMBDA,F,IDIMF,PERTRB,IERROR,W)
```

is replaced by

```
CALL HSTCRT

(A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,

+
ELMBDA,F,IDIMF,PERTRB,IERROR)
```

in FISHPACK90. All other arguments are identical. Workspace requirements in these solvers are transparent to the user. They are handled internally using pointers and dynamic array allocation.

The remaining five solvers

```
blktri.f,cblktri.f,hstcsp.f,hwscsp.f,se peli.f
```

have initial and non-initial calls utilizing saved workspace to reduce computational cost. These solvers require that the first declarative statement in the user program calling them is

```
USE fish
```

The user program should also include the declarative statement

```
TYPE (fishworkspace) :: w
```

The test programs for these solvers illustrate this. For example, look at the declarative statements in the test program file "tsepeli.f." Although the meaning of the USE and TYPE statements is transparent to the user, they do two things: they make the module "fish" available, and they declare a derived data type defined in "fish" that is used to allocate and pass real and complex workspace to lower-level subroutines. With any of these five solvers, users should also include the statement

CALL FISHFIN(W)

upon completion. This will de-allocate the saved workspace when it is no longer required. Failure to include this statement could result in serious memory leakage. This is also illustrated in the test programs for any of these five solvers.

Finally, all of the test programs for the 19 solvers have been rewritten using format-free I/O with results presented in 32-bit and 64-bit floating-point arithmetic. Documentation has been updated in all FISHPACK90 files to reflect the changes described.

Return to beginning of this document

Text Below Contains Internal Files Referenced by Above Links

BLKTRI

```
file blktri.txt (documentation for the FISHPACK
С
solver BLKTRI)
С
                        copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
С
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
*
С
```

```
С
                                       by
С
С
                John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                       of
С
      *
                 the National Center for Atmospheric
Research
С
                        Boulder, Colorado (80307)
U.S.A.
С
С
                            which is sponsored by
С
*
С
                      the National Science Foundation
С
С
      SUBROUTINE BLKTRI
(IFLG, NP, N, AN, BN, CN, MP, M, AM, BM, CM, IDIMY, Y,
С
                           IERROR, W)
С
С
С
C DIMENSION OF
AN (N), BN (N), CN (N), AM (M), BM (M), CM (M), Y (IDIMY, N),
C ARGUMENTS
С
C LATEST REVISION JUNE 2004
```

```
C USAGE
                          CALL BLKTRI
(IFLG, NP, N, AN, BN, CN, MP, M, AM, BM,
CM, IDIMY, Y, IERROR, W)
C PURPOSE
                          BLKTRI SOLVES A SYSTEM OF
LINEAR EQUATIONS
                          OF THE FORM
С
С
                          AN(J) *X(I,J-1) + AM(I) *X(I-1,J)
+
С
                          (BN(J)+BM(I))*X(I,J) +
CN(J) *X(I, J+1) +
                          CM(I) *X(I+1,J) = Y(I,J)
C
С
С
                          FOR I = 1, 2, \ldots, M AND J =
1, 2, \ldots, N.
C
                          I+1 AND I-1 ARE EVALUATED
MODULO M AND
С
                          J+1 AND J-1 MODULO N, I.E.,
С
C
                          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).
С
                          THESE EQUATIONS USUALLY RESULT
FROM THE
                          DISCRETIZATION OF SEPARABLE
ELLIPTIC
                          EOUATIONS. BOUNDARY CONDITIONS
MAY BE
                          DIRICHLET, NEUMANN, OR
PERIODIC.
С
C ARGUMENTS
C ON INPUT
                         IFLG
С
C
                            = 0 INITIALIZATION ONLY.
С
                                  CERTAIN OUANTITIES THAT
DEPEND ON NP,
                                 N, AN, BN, AND CN ARE
```

COMPUTED AND		
С		STORED IN DERIVED data
type w (see		
С		description of w below)
C		
C	= 1	THE QUANTITIES THAT WERE
COMPUTED		
C		IN THE INITIALIZATION
ARE USED		
C		TO OBTAIN THE SOLUTION
X(I,J).		
С		
С		NOTE:
C		A CALL WITH IFLG=0 TAKES
C		APPROXIMATELY ONE HALF
THE TIME		
С		AS A CALL WITH IFLG = 1 .
C		HOWEVER, THE
INITIALIZATION DOES		
C		NOT HAVE TO BE REPEATED
UNLESS NP,		
C		N, AN, BN, OR CN CHANGE.
С		
С	NP	
С	= 0	IF AN(1) AND CN(N) ARE
NOT ZERO,		
C		WHICH CORRESPONDS TO
PERIODIC		
С		BOUNARY CONDITIONS.
С		
С	= 1	IF AN(1) AND CN(N) ARE
ZERO.		, , , , , , , , , , , , , , , , , , , ,
C		
C	N	
C		NUMBER OF UNKNOWNS IN THE
J-DIRECTION.		
C	N MU	ST BE GREATER THAN 4.
C		OPERATION COUNT IS
PROPORTIONAL TO		
C	MNT.O	G2(N), HENCE N SHOULD BE
SELECTED	11110	
C	LESS	THAN OR EQUAL TO M.
C		11111 OI IQUIII 10 II.
C	AN, BN,	CN
	TAIN DIN	OTA .

С	ONE-DIMENSIONAL ARRAYS OF
LENGTH N	
C	THAT SPECIFY THE COEFFICIENTS
IN THE	
C	LINEAR EQUATIONS GIVEN ABOVE.
C	
C	MP
C	$= 0 ext{ IF AM (1)} ext{ AND CM (M)} ext{ ARE}$
NOT ZERO,	
C	WHICH CORRESPONDS TO
PERIODIC	
C	BOUNDARY CONDITIONS.
C	
C	$= 1 ext{ IF AM (1)} = CM (M) = 0 .$
C	
C	M
C	THE NUMBER OF UNKNOWNS IN THE
I-DIRECTION.	
C	M MUST BE GREATER THAN 4.
C	
C	AM, BM, CM
C	ONE-DIMENSIONAL ARRAYS OF
LENGTH M THAT	
C	SPECIFY THE COEFFICIENTS IN
THE LINEAR	
C	EQUATIONS GIVEN ABOVE.
C	TDTMY
C	THE ROW (OR FIRST) DIMENSION
OF THE	THE ROW (OR FIRST) DIMENSION
C	TWO-DIMENSIONAL ARRAY Y AS IT
APPEARS	IWO DIPENSIONAL ARVAI I AS II
C	IN THE PROGRAM CALLING
BLKTRI.	IN THE PROOFER CALLING
C C	THIS PARAMETER IS USED TO
SPECIFY THE	11110 11111111111 10 0010 10
C	VARIABLE DIMENSION OF Y.
C	IDIMY MUST BE AT LEAST M.
C	
C	Y
C	A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES	
C	THE VALUES OF THE RIGHT SIDE
OF THE LINEAR	

C	CVOMEN OF FOLIAMIONS CIVEN
C ABOVE.	SYSTEM OF EQUATIONS GIVEN
C C	Y MUST BE DIMENSIONED AT
LEAST M*N.	I MOSI DE DIMENSIONED AI
C	
C	
C	A fortran 90 derived TYPE
(fishworkspace) variable	
C	that must be declared by the
user. The first	-
С	declarative statement in the
user program	
С	calling BLKTRI must be:
C	
С	USE fish
C	
C	Additionally the declarative
statement	
C	TYPE (fishworkspace) ::
C	iirt (iisiiwoikspace)
C	
C	must also be included in the
user program.	
C	The first statement makes the
fishpack module	
С	defined in the file "fish.f"
available to the	
С	user program calling BLKTRI.
The second statement	
C	declares a derived type
variable (defined in	the medule Ufich EU) which is
C	the module "fish.f") which is
used internally c	in BLKTRI to dynamically
allocate real and complex	III DIIKIKI CO GYNGMICGILY
C C	work space used in solution.
An error flag	worm space used in serucion.
C	(IERROR = 20) is set if the
required work space	
С	allocation fails (for example
if N,M are too large)	
C	Real and complex values are
set in the components	

С	of W on a initial (IFLG=0)
call to BLKTRI. These	
c initial calls (IFLG=1)	must be preserved on non-
C C	to BLKTRI. This eliminates
redundant calculations	
C ****	and saves compute time.
c **** calling BLKTRI should	IMPORTANT! The user program
C C	include the statement:
С	
С	CALL FISHFIN(W)
C	after the final approximation
is generated by	arter the illiar approximation
C	BLKTRI. The will deallocate
the real and complex	
c include this statement	work space of W. Failure to
C C	could result in serious
memory leakage.	
С	
C C ARGUMENTS	
C ARGOMENTS	
C ON OUTPUT	Y
C	CONTAINS THE SOLUTION X.
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID	
C	INPUT PARAMETERS. EXCEPT FOR
NUMBER ZERO, C	A SOLUTION IS NOT ATTEMPTED.
C	A SOLUTION IS NOT ATTEMED.
С	= 0 NO ERROR.
С	= 1 M IS LESS THAN 5
C	= 2 N IS LESS THAN 5 = 3 IDIMY IS LESS THAN M.
C	= 4 BLKTRI FAILED WHILE
COMPUTING RESULTS	
C	THAT DEPEND ON THE
COEFFICIENT ARRAYS	
C C	AN, BN, CN. CHECK THESE

ARRAYS.				
C . FOR GOVER I	=	5	AN ((J)*CN(J-1) IS LESS THAN
0 FOR SOME J.				
C			POS	SSIBLE REASONS FOR THIS
CONDITION ARE				
С			1.	THE ARRAYS AN AND CN
ARE NOT CORRECT				
C			2.	TOO LARGE A GRID
SPACING WAS USED				TNI MIID DICCODUMICAMIONI
OF THE ELLIPTIC				IN THE DISCRETIZATION
C C				EQUATION.
C				THE LINEAR EQUATIONS
RESULTED FROM A				~
С				PARTIAL DIFFERENTIAL
EQUATION WHICH				
C				WAS NOT ELLIPTIC.
C	_	20	Т£	the dynamic allocation
of real and		20		the dynamic arrocation
C			con	plex work space in the
derived type				_
C			(fi	shworkspace) variable W
fails (e.g.,				
C			if	N,M are too large for
the platform used)				
C				
C	W			
C			The	e derived type
(fishworkspace) variable	M			
C			con	ntains real and complex
values that must not			,	1 ' 6 DITTED T '
C called again with			be	destroyed if BLKTRI is
called again with C			TTT	G=1.
C				1.
C				
C SPECIAL CONDITIONS	TH	IE P	ALGC	RITHM MAY FAIL IF
ABS(BM(I)+BN(J))				
C	IS	S LE	ESS	THAN
ABS (AM(I)) + ABS (AN(J)) +	7	00.46	71/1/ -	1) LADG (CM (T))
С	AE	35 ((()) +ABS (CN (J))

С	FOR SOME I AND J. THE ALGORITHM
WILL ALSO	
C	FAIL IF AN(J)*CN(J-1) IS LESS
THAN ZERO FOR	
C	SOME J.
C	SEE THE DESCRIPTION OF THE
OUTPUT PARAMETER	
C	IERROR.
C	
C I/O	NONE
C	
C PRECISION	SINGLE
C	
C REQUIRED FILES	fish.f,comf.f
С	
C LANGUAGE	FORTRAN 90
C	
C HISTORY	WRITTEN BY PAUL SWARZTRAUBER AT
NCAR IN THE	
C	EARLY 1970'S. REWRITTEN AND
RELEASED IN	
C	LIBRARIES IN JANUARY 1980.
Revised in June	
C	2004 using Fortan 90
dynamically allocated wo	
C	space and derived data types to
eliminate mixed	
C	mode conflicts in the earlier
versions.	
C	
C ALGORITHM	GENERALIZED CYCLIC REDUCTION
C	
C PORTABILITY	FORTRAN 90. APPROXIMATE
MACHINE ACCURACY	
C	IS COMPUTED IN FUNCTION EPMACH.
C	
C REFERENCES	SWARZTRAUBER, P. AND R. SWEET,
'EFFICIENT	HODERAN GURBERGERANG TOO THE
C	FORTRAN SUBPROGRAMS FOR THE
SOLUTION OF	
C	ELLIPTIC EQUATIONS'
C	NCAR TN/IA-109, JULY, 1975, 138
PP.	
C	

CBLKTRI

```
file cblktri.txt (documentation for the FISHPACK
solver CBLKTRI)
* * * * * * *
С
C
                         copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
С
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
*
С
С
                        A Package of Fortran 77 and 90
```

```
Subroutines and Example Programs
С
С
                       for Modeling Geophysical Processes
С
С
С
                                      by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                      of
С
С
                 the National Center for Atmospheric
Research
                        Boulder, Colorado (80307)
С
U.S.A.
                           which is sponsored by
С
С
                      the National Science Foundation
С
С
      SUBROUTINE CBLKTR
(IFLG, NP, N, AN, BN, CN, MP, M, AM, BM, CM, IDIMY, Y,
                          IERROR)
```

```
C
С
C DIMENSION OF
AN (N), BN (N), CN (N), AM (M), BM (M), CM (M), Y (IDIMY, N)
C ARGUMENTS
C LATEST REVISION
                          JUNE 2004
C PURPOSE
                          CBLKTR SOLVES A SYSTEM OF
LINEAR EQUATIONS
                          OF THE FORM
С
С
                          AN(J) *X(I,J-1) + AM(I) *X(I-1,J)
+
С
                           (BN(J)+BM(I))*X(I,J) +
CN(J) *X(I, J+1) +
С
                          CM(I) *X(I+1,J) = Y(I,J)
С
                          FOR I = 1, 2, \ldots, M AND J =
C
1, 2, \ldots, N.
С
                          I+1 AND I-1 ARE EVALUATED
C
MODULO M AND
                          J+1 AND J-1 MODULO N, I.E.,
C
С
C
                          X(I, 0) = X(I, N), X(I, N+1) =
X(I,1),
                          X(0,J) = X(M,J), X(M+1,J) =
С
X(1,J).
С
C.
                          THESE EQUATIONS USUALLY RESULT
FROM THE
                          DISCRETIZATION OF SEPARABLE
С
ELLIPTIC
                          EQUATIONS. BOUNDARY CONDITIONS
MAY BE
                          DIRICHLET, NEUMANN, OR
PERIODIC.
С
                          CBLKTRI IS A COMPLEX VERSION OF
C
PACKAGE
                          BLKTRI ON ULIB.
C
C
C USAGE
                          CALL CBLKTR
```

(IFLG, NP, N, AN, BN, CN, MP, M	I,AM,BM,	
C		
CM, IDIMY, Y, IERROR, W)		
C		
C ARGUMENTS		
C		
C ON INPUT	IFLG	
C	1110	
C	= 0	INITIALIZATION ONLY.
	- 0	
C		CERTAIN QUANTITIES THAT
DEPEND ON NP,		
C		N, AN, BN, AND CN ARE
COMPUTED AND		
C		STORED IN THE DERIVED
DATA TYPE W		
C		
С	= 1	THE QUANTITIES THAT WERE
COMPUTED		~
C		IN THE INITIALIZATION
ARE USED		
C		TO OBTAIN THE SOLUTION
		10 OBTAIN THE SOLUTION
X(I,J).		
C		
C		NOTE:
C		A CALL WITH IFLG=0 TAKES
C		APPROXIMATELY ONE HALF
THE TIME		
C		AS A CALL WITH IFLG $= 1$.
С		HOWEVER, THE
INITIALIZATION DOES		
C		NOT HAVE TO BE REPEATED
UNLESS NP,		
C		N, AN, BN, OR CN CHANGE.
C		II, AII, DII, OIL CII CIIAIIOE.
C	NID	
	NP	TO 351/1\ 3510 (\$1/51\ 300
C	= 0	IF AN(1) AND CN(N) ARE
NOT ZERO,		
C		WHICH CORRESPONDS TO
PERIODIC		
C		BOUNARY CONDITIONS.
С		
C	= 1	IF AN(1) AND CN(N) ARE
ZERO.		
С		

C N THE NUMBER OF UNKNOWNS	ב דאו ייטי
J-DIRECTION.	
C N MUST BE GREATER THAN	J 4.
C THE OPERATION COUNT IS	
PROPORTIONAL TO	
	NII D DE
	JOHO DE
SELECTED	D. (f.
C LESS THAN OR EQUAL TO	М.
C	
C AN, BN, CN	
C ONE-DIMENSIONAL ARRAYS	SOF
LENGTH N	
C THAT SPECIFY THE COEFI	FICIENTS
IN THE	
C LINEAR EQUATIONS GIVEN	N ABOVE.
C	
C MP	
C = 0 IF AM(1) AND CM(N)	1) ARE
NOT ZERO,	-,
C WHICH CORRESPONDS	з то
PERIODIC	5 10
C BOUNDARY CONDITION	MIC
C BOONDARI CONDITIO	JND.
	_ 0
C = 1 IF AM(1) = CM(M)	- 0 .
C	
C M	~
C THE NUMBER OF UNKNOWNS	S IN THE
I-DIRECTION.	
C M MUST BE GREATER THA	AN 4.
C	
C AM, BM, CM	
C COMPLEX ONE-DIMENSIONA	ĄL
ARRAYS OF LENGTH M	
C THAT SPECIFY THE COEFF	FICIENTS
IN THE LINEAR	
C EQUATIONS GIVEN ABOVE.	
C	
C IDIMY	
C THE ROW (OR FIRST) DIN	ÆNSTON
OF THE	V OIV
C TWO-DIMENSIONAL ARRAY	V NC TM
	1 AO II
APPEARS IN THE PROGRAM CALLING	7
C IN THE PROGRAM CALLING	J

CBLKTR.	
C	THIS PARAMETER IS USED TO
SPECIFY THE	
C	VARIABLE DIMENSION OF Y.
C	IDIMY MUST BE AT LEAST M.
C	,
C	
C	A COMPLEX TWO-DIMENSIONAL
ARRAY THAT	
C DIGITAL CIDE OF	SPECIFIES THE VALUES OF THE
RIGHT SIDE OF	MIE I INEAD CYCMEM OF
C EQUIDATIONS CIVEN ADOME	THE LINEAR SYSTEM OF
EQUATIONS GIVEN ABOVE.	Y MUST BE DIMENSIONED
	I MOSI BE DIMENSIONED
Y(IDIMY,N) WITH C	IDIMY .GE. M.
C	IDIMI .GE. M.
C	7
C	A fortran 90 derived TYPE
(fishworkspace) variable	A TOTCTAIL 90 GETTVEG TITE
C (IISHWOIKSPACE) VALIABLE	that must be declared by the
user. The first	that must be declared by the
C C	two declarative statements in
the user program	ewo decidiative beatements in
C	calling CBLKTRI must be:
C	outing oblining must be.
C	USE fish
C	002 2201
C	Additionally the declarative
statement	
C	
C	TYPE (fishworkspace) ::
W	,
C	
C	The first statement makes the
fishpack module	
C	defined in the file "fish.f"
available to the	
C	user program calling CBLKTRI.
The second statement	
С	declares a derived type
variable (defined in	
С	the module "fish.f") which is
used internally	

c allocate real and complex	in CBLKTRI to dynamically
c An error flag	work space used in solution.
С	(IERROR = 20) is set if the
required work space c if N,M are too large)	allocation fails (for example
c set in the components	Real and complex values are
C	of W on a initial (IFLG=0)
call to CBLKTRI. These	must be preserved on non-
initial calls (IFLG=1)	to CBLKTRI. This eliminates
redundant calculations c c ****	and saves compute time. IMPORTANT! The user program
calling CBLKTRI should c	include the statement:
C	CALL FISHFIN(W)
C	after the final approximation
is generated by C	CBLKTRI. The will deallocate
the real and complex c	work space of W. Failure to
include this statement	could result in serious
memory leakage.	
C	
C ARGUMENTS C	
C ON OUTPUT	Y CONTAINS THE SOLUTION X.
C :	IERROR
C INVALID	AN ERROR FLAG THAT INDICATES
C	INPUT PARAMETERS. EXCEPT FOR
NUMBER ZERO, C	A SOLUTION IS NOT ATTEMPTED.

C	
C	= 0 NO ERROR.
C	= 1 M IS LESS THAN 5
C	= 2 N IS LESS THAN 5
C	= 3 IDIMY IS LESS THAN M.
C	= 4 CBLKTR FAILED WHILE
COMPUTING RESULTS	
C	THAT DEPEND ON THE
COEFFICIENT ARRAYS	
C	AN, BN, CN. CHECK THESE
ARRAYS.	, ,
C	= 5 AN (J) *CN (J-1) IS LESS THAN
0 FOR SOME J.	5 74V(O) CIV(O I) IS LLESS III.AV
C	
C	POSSIBLE REASONS FOR THIS
CONDITION ARE	
С	1. THE ARRAYS AN AND CN
ARE NOT CORRECT	
C	2. TOO LARGE A GRID
SPACING WAS USED	1, 100 ====0 == ===
C C	IN THE DISCRETIZATION
	IN THE DISCRETIZATION
OF THE ELLIPTIC	
C	EQUATION.
C	3. THE LINEAR EQUATIONS
RESULTED FROM A	
C	PARTIAL DIFFERENTIAL
EQUATION WHICH	
C	WAS NOT ELLIPTIC.
C	WIND INCI LIBERTIES.
	- 20 Tf the demands
C	= 20 If the dynamic
allocation of real and	
C	complex work space in
the derived type	
C	(fishworkspace) variable
W fails (e.g.,	-
C (Style)	if N,M are too large for
the platform used)	11 11,11 a10 000 1a190 101
_	
C	
C	
C	
C SPECIAL CONDITIONS	THE ALGORITHM MAY FAIL IF
ABS(BM(I)+BN(J))	
C	IS LESS THAN
ABS (AM(I)) + ABS (AN(J)) +	
120 (121(1)) 1120 (121(0))	

C	ABS(CM(I))+ABS(CN(J))
С	FOR SOME I AND J. THE ALGORITHM
WILL ALSO	
С	FAIL IF $AN(J)*CN(J-1)$ IS LESS
THAN ZERO FOR	
C	SOME J.
C	SEE THE DESCRIPTION OF THE
OUTPUT PARAMETER	
C	IERROR.
C	
C	
C I/O	NONE
C	
C PRECISION	SINGLE
C	
C REQUIRED LIBRARY	comf.f, fish.f
C FILES	
C	TOPEDAN. OO
C LANGUAGE	FORTRAN 90
С	
C HISTORY	WRITTEN BY PAUL SWARZTRAUBER AT
NCAR IN	MIE EADIV 1070 C DEMOTMENI ANI
C RELEASED	THE EARLY 1970'S. REWRITTEN AN
C	ON NCAR'S PUBLIC SOFTWARE
LIBRARIES IN	ON NCAR 5 FUBLIC SUFIWARE
C	JANUARY, 1980. Revised in June
2004 by John	OANOAKI, 1900. Kevised in oune
C	Adams using Fortan 90
dynamically allocated	ridanis astrig refeati 50
C	space and derived data types to
eliminate mixed	space and delived dates offer of
C	mode conflicts in the earlier
versions.	
С	
C ALGORITHM	GENERALIZED CYCLIC REDUCTION
С	(SEE REFERENCE BELOW)
С	,
C PORTABILITY	
C	THE APPROXIMATE MACHINE
ACCURACY IS COMPUTED	
С	IN FUNCTION EPMACH
С	
C REFERENCES	SWARZTRAUBER, P. AND R. SWEET,

```
'EFFICIENT
C
                    FORTRAN SUBPROGRAMS FOR THE
SOLUTION OF
                    ELLIPTIC EOUATIONS'
                    NCAR TN/IA-109, JULY, 1975, 138
С
PP.
C
С
                    SWARZTRAUBER P. N., A DIRECT
METHOD FOR
                    THE DISCRETE SOLUTION OF
SEPARABLE
                    ELLIPTIC EQUATIONS, S.I.A.M.
                    J. NUMER. ANAL., 11 (1974) PP.
1136-1150.
*****
```

CMGNBN

```
FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
С
                     the National Science Foundation
```

```
C
С
      SUBROUTINE CMGNBN
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y, IERROR)
С
C DIMENSION OF
                          A(M), B(M), C(M), Y(IDIMY, N)
C ARGUMENTS
C LATEST REVISION
                         NOVEMBER 2004
C PURPOSE
                          THE NAME OF THIS PACKAGE IS A
MNEMONIC FOR THE
                          COMPLEX GENERALIZED BUNEMAN
ALGORITHM.
                          IT SOLVES THE COMPLEX LINEAR
SYSTEM OF EQUATION
C
                          A(I) * X(I-1,J) + B(I) * X(I,J) +
C(I) *X(I+1,J)
                          + X(I,J-1) - 2.*X(I,J) +
X(I,J+1) = Y(I,J)
C
                          FOR I = 1, 2, \ldots, M AND J =
1, 2, ..., N.
                          INDICES I+1 AND I-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 EQUAL 0, X(I,2),
OR X(I,N),
                          AND X(I,N+1) MAY EQUAL 0,
C
X(I,N-1), OR X(I,1)
                          DEPENDING ON AN INPUT
PARAMETER.
C
C USAGE
                          CALL CMGNBN
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y,
С
                                         IERROR)
С
```

```
C ARGUMENTS
C
C ON INPUT
                           NPEROD
C
С
                               INDICATES THE VALUES THAT
X(I,0) AND
                              X(I,N+1) ARE ASSUMED TO HAVE.
C
С
С
                              = 0 	 IF X(I,0) = X(I,N) 	 AND
X(I,N+1) =
                                    X(I,1).
С
                              = 1 	ext{ IF } X(I,0) = X(I,N+1) = 0
                              = 2 \quad \text{IF } X(I,0) = 0 \quad \text{AND}
X(I,N+1) = X(I,N-1).
                              = 3 	ext{ IF } X(I,0) = X(I,2) 	ext{ AND}
X(I,N+1) =
                                    X(I, N-1).
C
C
                              = 4 \quad \text{IF } X(I,0) = X(I,2) \quad \text{AND}
X(I,N+1) = 0.
С
С
                            N
C
                               THE NUMBER OF UNKNOWNS IN THE
J-DIRECTION.
                              N MUST BE GREATER THAN 2.
С
С
                            MPEROD
С
                              = 0 IF A(1) AND C(M) ARE NOT
ZERO
С
                              = 1 \text{ IF A (1)} = C(M) = 0
С
С
                            M
С
                              THE NUMBER OF UNKNOWNS IN THE
I-DIRECTION.
С
                              N MUST BE GREATER THAN 2.
С
С
                            A,B,C
С
                              ONE-DIMENSIONAL COMPLEX
ARRAYS OF LENGTH M
                              THAT SPECIFY THE COEFFICIENTS
IN THE LINEAR
                              EQUATIONS GIVEN ABOVE. IF
MPEROD = 0
                               THE ARRAY ELEMENTS MUST NOT
```

DEPEND UPON	
C	THE INDEX I, BUT MUST BE
CONSTANT.	
C	SPECIFICALLY, THE SUBROUTINE
CHECKS THE	
C	FOLLOWING CONDITION .
C	
С	A(I) = C(1)
С	C(I) = C(1)
С	B(I) = B(1)
С	
C	FOR $I=1,2,,M$.
C	
C	IDIMY
C	THE ROW (OR FIRST) DIMENSION
OF THE	
C	TWO-DIMENSIONAL ARRAY Y AS IT
APPEARS	
С	IN THE PROGRAM CALLING
CMGNBN.	
С	THIS PARAMETER IS USED TO
SPECIFY THE	
С	VARIABLE DIMENSION OF Y.
C	IDIMY MUST BE AT LEAST M.
C	
C	Y
C	A TWO-DIMENSIONAL COMPLEX
ARRAY THAT	
C	SPECIFIES THE VALUES OF THE
RIGHT SIDE	
C	OF THE LINEAR SYSTEM OF
EQUATIONS GIVEN	
C	ABOVE.
C	Y MUST BE DIMENSIONED AT
LEAST M*N.	
C	
C	
C ON OUTPUT	Y
С	
С	CONTAINS THE SOLUTION X.
С	
С	IERROR
С	AN ERROR FLAG WHICH INDICATES
INVALID	

C	INPUT PARAMETERS EXCEPT FOR
NUMBER	EPPO A COLUMNOM TO NOT
C	ZERO, A SOLUTION IS NOT
ATTEMPTED.	
C	= 0 NO ERROR.
C	= 1 M .LE. 2 .
C	= 2 N .LE. 2
C	= 3 IDIMY .LT. M
C	= 4 NPEROD .LT. 0 OR NPEROD
.GT. 4	
C	= 5 MPEROD .LT. 0 OR MPEROD
.GT. 1	
C	= 6 A(I) .NE. C(I) OR C(I)
.NE. C(1) OR	
C	B(I) .NE. $B(1)$ FOR
С	SOME $I=1,2,\ldots,M$.
С	= 7 A(1) .NE. 0 OR C(M) .NE.
0 AND	
С	MPEROD = 1
С	= 20 If the dynamic
allocation of real and	
С	complex work space
required for solution	
С	fails (for example if
N,M are too large	
C	for your computer)
C	
	NONE
C	
C I/O	NONE
C	CTNCL F
C PRECISION	SINGLE
C DECLIDED LIDDADY	anne e elab e
C REQUIRED LIBRARY	comf.f, fish.f
C FILES	
C LANGUAGE	FORTRAN 90
C LANGUAGE	FOLTIVAN 90
C HISTORY	WRITTEN IN 1979 BY ROLAND SWEET
OF NCAR'S	MINITITIES THE TOLO DI LOTURE OMPETE
C	SCIENTIFIC COMPUTING DIVISION.
MADE AVAILABLE	COLLINITIE CONTROLLING DIVIDION.
C	ON NCAR'S PUBLIC LIBRARIES IN
-	

JANUARY, 1980.	
С	Revised in June 2004 by John
Adams using	
С	Fortran 90 dynamically
allocated work space.	
C	
C ALGORITHM	THE LINEAR SYSTEM IS SOLVED BY
A CYCLIC	
C	REDUCTION ALGORITHM DESCRIBED
IN THE	
C	REFERENCE BELOW.
C	
C PORTABILITY	FORTRAN 90. ALL MACHINE
DEPENDENT CONSTANTS	
C	ARE DEFINED IN FUNCTION P1MACH.
C	
C REFERENCES	SWEET, R., 'A CYCLIC REDUCTION
ALGORITHM FOR	
C	SOLVING BLOCK TRIDIAGONAL
SYSTEMS OF ARBITRARY	
C	DIMENSIONS, 'SIAM J. ON NUMER.
ANAL.,	
C	14(SEPT., 1977), PP. 706-720.
C	
C ACCURACY	THIS TEST WAS PERFORMED ON A
Platform with	
C	64 bit floating point
arithmetic.	
C	A UNIFORM RANDOM NUMBER
GENERATOR WAS USED	
C	TO CREATE A SOLUTION ARRAY X
FOR THE SYSTEM	
C	GIVEN IN THE 'PURPOSE'
DESCRIPTION ABOVE	
C	WITH
C	A(I) = C(I) = -0.5*B(I) = 1,
I=1,2,,M	
C	AND DUIDNI MORDOD 1
C	AND, WHEN MPEROD = 1
C	7/1 $- C/M$ $- 0$
C	A(1) = C(M) = 0
C	A(M) = C(1) = 2.
C	MILE COLLIMIAN A FIX C OLLD CHILDREN
C	THE SOLUTION X WAS SUBSTITUTED

INTO THE			711D 7 DTG	NIE GEDE
C	G.	IVEN SYSTEM	AND A RIG	HT SIDE
Y WAS				
C	C	OMPUTED. USI	NG THIS A	ARRAY Y,
SUBROUTINE				
С	CI	MGNBN WAS CAI	LED TO PF	RODUCE
APPROXIMATE				
С	S	DLUTION Z. I	HEN RELAT	IVE
ERROR				
C		E = MAX (CABS)	S(Z(I,J)-X)	X(I,J)))/
С		MAX (CABS	S(X(I,J))	
С	W	AS COMPUTED,	WHERE THE	I TWO
MAXIMA ARE TAKEN				
С	O _z	/ER I=1,2,	,M AND J=	=1,,N.
C				
C	T	HE VALUE OF E	IS GIVEN	I IN THE
TABLE				
С	В	ELOW FOR SOME	TYPICAL	VALUES
OF M AND N.				
С				
С	M (=N)	MPEROD	NPEROD	E
C				
_				
С				
C	31	0	0	1.E-
12	01	O	O	1.5
C	31	1	1	4.E-
13	91	_	_	1 • 1
C	31	1	3	2.E-
12	JI	±	5	2 • 11
C	32	0	0	7.E-
14	52	O	O	/ • <u>ii</u> —
C	20	1	1	5 17
	32	1	1	5.E-
13	20	1	2	0 11
C 12	32	1	3	2.E-
13	22	0	0	6.5
C	33	0	0	6.E-
13	0.0	4	1	
C	33	1	1	5.E-
13				
C	33	1	3	3.E-
12				
C	63	0	0	5.E-
12				

63	1	1	6.E-
63	1	3	1.E-
64	0	0	1.E-
64	1	1	3.E-
64	1	3	3.E-
65	0	0	2.E-
65	1	1	5.E-
65	1	3	1.E-
C****************			
	63646464656565	63 1 0 0 64 1 65 0 65 1 65 1	63 1 3 64 0 0 64 1 1 64 1 3 65 0 0 65 1 1 65 1 3

COMF

```
all rights reserved
С
C
                            FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
С
                      for Modeling Geophysical Processes
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
                                     of
С
                the National Center for Atmospheric
С
Research
                       Boulder, Colorado (80307)
С
U.S.A.
                          which is sponsored by
С
С
```

C *	the National Science Foundation
C *	
	* * * * * * * * * * * * * * *
C * * * * * * * * *	
C	
C PACKAGE COMF	THE ENTRIES IN THIS PACKAGE ARE
LOWLEVEL	THE ENTRIES IN THIS PACKAGE ARE
C	ENTRIES, SUPPORTING FISHPACK
ENTRIES BLKTRI	ENTINES, SOLIONIENG FISHLACK
C DERVINE	AND CBLKTRI. THAT IS, THESE
ROUTINES ARE	THE COURTER TO THE TOP
C	NOT CALLED DIRECTLY BY USERS,
BUT RATHER	NOT CHILLED DITUCTION DI COLINO,
C	BY ENTRIES WITHIN BLKTRI AND
CBLKTRI.	
С	DESCRIPTION OF ENTRIES EPMACH
AND PIMACH	
С	FOLLOW BELOW.
С	
C LATEST REVISION	JUNE 2004
С	
C SPECIAL CONDITIONS	NONE
С	
C I/O	NONE
С	
C PRECISION	SINGLE
С	
C REQUIRED LIBRARY	NONE
C FILES	
C	TODED AND OO
C LANGUAGE	FORTRAN 90
C	*********
*****	^^^^^
C	
C FUNCTION EPMACH (DU	M)
C FONCTION EFMACII (DO	/
C PURPOSE	TO COMPUTE AN APPROXIMATE
MACHINE ACCURACY	
C	EPSILON ACCORDING TO THE
FOLLOWING DEFINITION:	

C	EPSILON IS THE SMALLEST NUMBER
SUCH THAT	
C	(1.+EPSILON).GT.1.)
C	
C USAGE	EPS = EPMACH (DUM)
C	
C ARGUMENTS	
C ON INPUT	DUM
C	DUMMY VALUE
C	
C ARGUMENTS	
C ON OUTPUT	NONE
C	
C HISTORY	THE ORIGINAL VERSION, WRITTEN
WHEN THE	
С	BLKTRI PACKAGE WAS CONVERTED
FROM THE	
С	CDC 7600 TO RUN ON THE CRAY-1,
CALCULATED	
С	MACHINE ACCURACY BY SUCCESSIVE
DIVISIONS	
С	BY 10. USE OF THIS CONSTANT
CAUSED BLKTRI	
С	TO COMPUTE SOLUTIONS ON THE
CRAY-1 WITH FOUR	
С	FEWER PLACES OF ACCURACY THAN
THE VERSION	
С	ON THE 7600. IT WAS FOUND THAT
COMPUTING	
С	MACHINE ACCURACY BY SUCCESSIVE
DIVISIONS	
С	OF 2 PRODUCED A MACHINE
ACCURACY 29% LESS	
С	THAN THE VALUE GENERATED BY
SUCCESSIVE	
С	DIVISIONS BY 10, AND THAT USE
OF THIS	·
С	MACHINE CONSTANT IN THE BLKTRI
PACKAGE	
С	RECOVERED THE ACCURACY THAT
APPEARED TO	
C	BE LOST ON CONVERSION.
C	
C ALGORITHM	COMPUTES MACHINE ACCURACY BY

```
SUCCESSIVE
C
                    DIVISIONS OF TWO.
C
C PORTABILITY
                    THIS CODE WILL EXECUTE ON
MACHINES OTHER
                    THAN THE CRAY1, BUT THE
RETURNED VALUE MAY
                    BE UNSATISFACTORY. SEE HISTORY
ABOVE.
***************
*****
C FUNCTION PIMACH (DUM)
C PURPOSE
                TO SUPPLY THE VALUE OF THE
CONSTANT PI
С
                   CORRECT TO MACHINE PRECISION
WHERE
PI=3.141592653589793238462643383279502884197
                        1693993751058209749446
C
C USAGE
                   PI = PIMACH (DUM)
C ARGUMENTS
C ON INPUT
                    DUM
С
                     DUMMY VALUE
С
C ARGUMENTS
C ON OUTPUT
                   NONE
C
C ALGORITHM
                    THE VALUE OF PI IS SET TO
4.*ATAN(1.0)
C PORTABILITY
              THIS ENTRY IS PORTABLE, BUT
USERS SHOULD
                    CHECK TO SEE WHETHER GREATER
ACCURACY IS
C
                    REQUIRED.
*****
```

FFTPACK

```
A Package of Fortran 77 and 90
С
                        Subroutines and Example Programs
C
С
                       for Modeling Geophysical Processes
C
C
С
                                      by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
\mathsf{C}
С
                                      of
С
                 the National Center for Atmospheric
Research
                        Boulder, Colorado (80307)
U.S.A.
С
                           which is sponsored by
С
С
C
                      the National Science Foundation
С
C
C LATEST REVISION
```

C			
С	June	2004	(VERSION 5.0) FORTRAN 90 CHANGES
С			
C PU	RPOSE		
C			
С	THIS	PACKAGE C	CONSISTS OF PROGRAMS WHICH PERFORM
FAST	FOURI	ER	
С	TRAN	SFORMS FOR	R BOTH COMPLEX AND REAL PERIODIC
SEQU:	ENCES .	AND	
С	CERT.	AIN OTHER	SYMMETRIC SEQUENCES THAT ARE LISTED
BELO	W.		
С			
C US	AGE		
C			
С	1.	RFFTI	INITIALIZE RFFTF AND RFFTB
С	2.	RFFTF	FORWARD TRANSFORM OF A REAL
PERI	ODIC S	EQUENCE	
С	3.	RFFTB	BACKWARD TRANSFORM OF A REAL
COEF	FICIEN	T ARRAY	
С			
С	4.	EZFFTI	INITIALIZE EZFFTF AND EZFFTB
С	5.	EZFFTF	A SIMPLIFIED REAL PERIODIC FORWARD
TRAN	SFORM		
С	6.	EZFFTB	A SIMPLIFIED REAL PERIODIC BACKWARD
TRAN	SFORM		
С			
С	7.	SINTI	INITIALIZE SINT
		SINT	SINE TRANSFORM OF A REAL ODD
SEQU:	ENCE		
С			
		COSTI	
С		COST	COSINE TRANSFORM OF A REAL EVEN
SEQU	ENCE		
С			
			INITIALIZE SINQF AND SINQB
		SINQF	FORWARD SINE TRANSFORM WITH ODD
	NUMBE		
	13.	SINQB	UNNORMALIZED INVERSE OF SINQF
С			
			INITIALIZE COSQF AND COSQB
		COSQF	FORWARD COSINE TRANSFORM WITH ODD
	NUMBE		
	16.	COSQB	UNNORMALIZED INVERSE OF COSQF
С			

```
C 17. CFFTI INITIALIZE CFFTF AND CFFTB
C 18. CFFTF FORWARD TRANSFORM OF A COMPLEX
PERIODIC SEQUENCE
C 19. CFFTB UNNORMALIZED INVERSE OF CFFTF
С
C SPECIAL CONDITIONS
C BEFORE CALLING ROUTINES EZFFTB AND EZFFTF FOR THE
FIRST TIME,
     OR BEFORE CALLING EZFFTB AND EZFFTF WITH A
DIFFERENT LENGTH,
C USERS MUST INITIALIZE BY CALLING ROUTINE EZFFTI.
С
C I/O
C ---
C NONE
C PRECISION
C. ----
C NONE
С
C REQUIRED LIBRARY FILES
C -----
C NONE
С
C LANGUAGE
C ----
C FORTRAN 90
С
C HISTORY
C. ----
    DEVELOPED AT NCAR IN BOULDER, COLORADO BY PAUL N.
SWARZTRAUBER
    OF THE SCIENTIFIC COMPUTING DIVISION. RELEASED ON
NCAR'S PUBLIC
    SOFTWARE LIBRARIES IN JANUARY 1980. MODIFIED MAY
29, 1985 TO
    INCREASE EFFICIENCY. Fortran 90 changes made June
2004
С
C PORTABILITY
C -----
C FORTRAN 90
```

```
********************
******
С
     SUBROUTINE RFFTI (N, WSAVE)
С
C
     SUBROUTINE RFFTI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     BOTH RFFTF AND RFFTB. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
    STORED IN WSAVE.
С
С
    INPUT PARAMETER
C
        THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED.
    OUTPUT PARAMETER
С
C
    WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 2*N+15.
            THE SAME WORK ARRAY CAN BE USED FOR BOTH
RFFTF AND RFFTB
            AS LONG AS N REMAINS UNCHANGED. DIFFERENT
WSAVE ARRAYS
            ARE REQUIRED FOR DIFFERENT VALUES OF N.
THE CONTENTS OF
            WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF
RFFTF OR RFFTB.
*************
С
     SUBROUTINE RFFTF (N, R, WSAVE)
C
С
     SUBROUTINE RFFTF COMPUTES THE FOURIER COEFFICIENTS
OF A REAL
     PERODIC SEQUENCE (FOURIER ANALYSIS). THE TRANSFORM
IS DEFINED
C
     BELOW AT OUTPUT PARAMETER R.
```

```
C INPUT PARAMETERS
С
             THE LENGTH OF THE ARRAY R TO BE
TRANSFORMED. THE METHOD
              IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
             N MAY CHANGE SO LONG AS DIFFERENT WORK
ARRAYS ARE PROVIDED
C
            A REAL ARRAY OF LENGTH N WHICH CONTAINS
     R
THE SEQUENCE
             TO BE TRANSFORMED
С
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 2*N+15.
              IN THE PROGRAM THAT CALLS RFFTF. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
RFFTI (N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
             REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSECUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
C
FIRST.
             THE SAME WSAVE ARRAY CAN BE USED BY RFFTF
C
AND RFFTB.
С
C
С
     OUTPUT PARAMETERS
С
     R = R(1) = THE SUM FROM I=1 TO I=N OF R(I)
С
С
C
              IF N IS EVEN SET L =N/2 , IF N IS ODD
SET L = (N+1)/2
C
С
                THEN FOR K = 2, ..., L
С
                   R(2*K-2) = THE SUM FROM I = 1 TO I =
C
N OF
С
                      R(I) *COS((K-1) * (I-1) *2*PI/N)
```

```
С
С
                  R(2*K-1) = THE SUM FROM I = 1 TO I =
N OF
С
С
                      -R(I)*SIN((K-1)*(I-1)*2*PI/N)
С
С
             IF N IS EVEN
С
С
                  R(N) = THE SUM FROM I = 1 TO I = N OF
С
С
                       (-1)**(I-1)*R(I)
С
С
      ****
            NOTE
                  THIS TRANSFORM IS UNNORMALIZED SINCE
A CALL OF RFFTF
C
                  FOLLOWED BY A CALL OF RFFTB WILL
MULTIPLY THE INPUT
C
                  SEQUENCE BY N.
С
     WSAVE CONTAINS RESULTS WHICH MUST NOT BE
DESTROYED BETWEEN
             CALLS OF RFFTF OR RFFTB.
C
С
****************
*****
С
     SUBROUTINE RFFTB (N, R, WSAVE)
С
     SUBROUTINE RFFTB COMPUTES THE REAL PERODIC
SEOUENCE FROM ITS
     FOURIER COEFFICIENTS (FOURIER SYNTHESIS). THE
TRANSFORM IS DEFINED
     BELOW AT OUTPUT PARAMETER R.
С
С
    INPUT PARAMETERS
С
             THE LENGTH OF THE ARRAY R TO BE
     N
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
             N MAY CHANGE SO LONG AS DIFFERENT WORK
ARRAYS ARE PROVIDED
```

```
С
         A REAL ARRAY OF LENGTH N WHICH CONTAINS
С
     R
THE SEQUENCE
C
              TO BE TRANSFORMED
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 2*N+15.
              IN THE PROGRAM THAT CALLS RFFTB. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
RFFTI (N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
              VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
              REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
              TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
              THE SAME WSAVE ARRAY CAN BE USED BY RFFTF
AND RFFTB.
C
С
     OUTPUT PARAMETERS
С
С
     R FOR N EVEN AND FOR I = 1, ..., N
С
С
                   R(I) = R(1) + (-1) ** (I-1) *R(N)
С
С
                        PLUS THE SUM FROM K=2 TO K=N/2
OF
С
                        2.*R(2*K-2)*COS((K-1)*(I-
С
1) *2*PI/N)
С
С
                        -2.*R(2*K-1)*SIN((K-1)*(I-
1) *2*PI/N)
С
С
              FOR N ODD AND FOR I = 1, ..., N
С
С
                   R(I) = R(1) PLUS THE SUM FROM K=2 TO
K = (N+1)/2 OF
С
                       2.*R(2*K-2)*COS((K-1)*(I-
```

```
1) *2*PI/N)
С
C
                     -2.*R(2*K-1)*SIN((K-1)*(I-
1) *2*PI/N)
C
      **** NOTE
C
                 THIS TRANSFORM IS UNNORMALIZED SINCE
C
A CALL OF RFFTF
                 FOLLOWED BY A CALL OF RFFTB WILL
MULTIPLY THE INPUT
                 SEOUENCE BY N.
С
C
    WSAVE CONTAINS RESULTS WHICH MUST NOT BE
DESTROYED BETWEEN
             CALLS OF RFFTB OR RFFTF.
С
С
С
*****************
*****
С
    SUBROUTINE EZFFTI (N, WSAVE)
C
     SUBROUTINE EZFFTI INITIALIZES THE ARRAY WSAVE
WHICH IS USED IN
     BOTH EZFFTF AND EZFFTB. THE PRIME FACTORIZATION OF
N TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
    STORED IN WSAVE.
С
C
С
    INPUT PARAMETER
С
    N THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED.
С
C OUTPUT PARAMETER
C
    WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             THE SAME WORK ARRAY CAN BE USED FOR BOTH
EZFFTF AND EZFFTB
            AS LONG AS N REMAINS UNCHANGED. DIFFERENT
WSAVE ARRAYS
```

```
C
             ARE REQUIRED FOR DIFFERENT VALUES OF N.
С
C
**************
******
С
     SUBROUTINE EZFFTF (N, R, AZERO, A, B, WSAVE)
С
     SUBROUTINE EZFFTF COMPUTES THE FOURIER
COEFFICIENTS OF A REAL
     PERODIC SEQUENCE (FOURIER ANALYSIS). THE TRANSFORM
IS DEFINED
     BELOW AT OUTPUT PARAMETERS AZERO, A AND B. EZFFTF
IS A SIMPLIFIED
    BUT SLOWER VERSION OF RFFTF.
С
С
    INPUT PARAMETERS
С
            THE LENGTH OF THE ARRAY R TO BE
     N
TRANSFORMED. THE METHOD
             IS MUST EFFICIENT WHEN N IS THE PRODUCT OF
SMALL PRIMES.
С
         A REAL ARRAY OF LENGTH N WHICH CONTAINS
    R
THE SEQUENCE
             TO BE TRANSFORMED. R IS NOT DESTROYED.
С
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             IN THE PROGRAM THAT CALLS EZFFTF. THE
WSAVE ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
EZFFTI (N, WSAVE) AND A
             DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
             REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
             THE SAME WSAVE ARRAY CAN BE USED BY EZFFTF
```

```
AND EZFFTB.
C
С
     OUTPUT PARAMETERS
C
С
     AZERO THE SUM FROM I=1 TO I=N OF R(I)/N
С
            FOR N EVEN B(N/2)=0. AND A(N/2) IS THE SUM
C
     A,B
FROM I=1 TO
С
             I=N OF (-1)**(I-1)*R(I)/N
С
С
             FOR N EVEN DEFINE KMAX=N/2-1
С
             FOR N ODD DEFINE KMAX = (N-1)/2
С
С
             THEN FOR K=1, \ldots, KMAX
С
С
                  A(K) EQUALS THE SUM FROM I=1 TO I=N
OF
С
С
                       2./N*R(I)*COS(K*(I-1)*2*PI/N)
С
С
                  B(K) EQUALS THE SUM FROM I=1 TO I=N
OF
С
С
                       2./N*R(I)*SIN(K*(I-1)*2*PI/N)
С
С
****************
*****
C
     SUBROUTINE EZFFTB (N, R, AZERO, A, B, WSAVE)
C
С
     SUBROUTINE EZFFTB COMPUTES A REAL PERODIC SEQUENCE
FROM ITS
     FOURIER COEFFICIENTS (FOURIER SYNTHESIS). THE
TRANSFORM IS
     DEFINED BELOW AT OUTPUT PARAMETER R. EZFFTB IS A
SIMPLIFIED
     BUT SLOWER VERSION OF RFFTB.
C
C
     INPUT PARAMETERS
С
C
         THE LENGTH OF THE OUTPUT ARRAY R.
                                               THE
METHOD IS MOST
```

```
C EFFICIENT WHEN N IS THE PRODUCT OF SMALL
PRIMES.
С
C
    AZERO THE CONSTANT FOURIER COEFFICIENT
С
    A,B ARRAYS WHICH CONTAIN THE REMAINING FOURIER
C
COEFFICIENTS
             THESE ARRAYS ARE NOT DESTROYED.
С
             THE LENGTH OF THESE ARRAYS DEPENDS ON
WHETHER N IS EVEN OR
             ODD.
С
С
             IF N IS EVEN N/2 LOCATIONS ARE REQUIRED
С
             IF N IS ODD (N-1)/2 LOCATIONS ARE REQUIRED
C
    WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             IN THE PROGRAM THAT CALLS EZFFTB. THE
WSAVE ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
EZFFTI (N, WSAVE) AND A
             DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
             REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
C.
             THE SAME WSAVE ARRAY CAN BE USED BY EZFFTF
AND EZFFTB.
С
С
С
     OUTPUT PARAMETERS
С
С
    R
             IF N IS EVEN DEFINE KMAX=N/2
С
             IF N IS ODD DEFINE KMAX = (N-1)/2
С
С
             THEN FOR I=1, ..., N
С
С
                  R(I)=AZERO PLUS THE SUM FROM K=1 TO
K=KMAX OF
```

```
A(K) *COS(K*(I-
1) *2*PI/N) +B(K) *SIN(K*(I-1)*2*PI/N)
C
     ***** COMPLEX NOTATION
С
             FOR J=1, \ldots, N
С
С
С
            R(J) EQUALS THE SUM FROM K=-KMAX TO K=KMAX
OF
С
C
                 C(K) *EXP(I*K*(J-1)*2*PI/N)
С
С
             WHERE
С
С
                 C(K) = .5*CMPLX(A(K), -B(K)) FOR
K=1, \ldots, KMAX
C
С
                 C(-K) = CONJG(C(K))
С
С
                 C(0) = AZERO
С
С
                      AND I=SORT(-1)
     ****** AMPLITUDE - PHASE NOTATION
*******
C
С
             FOR I=1, \ldots, N
С
С
             R(I) EQUALS AZERO PLUS THE SUM FROM K=1 TO
K=KMAX OF
C
                 ALPHA(K) *COS(K*(I-1) *2*PI/N+BETA(K))
С
С
С
             WHERE
С
С
                 ALPHA(K) = SQRT(A(K)*A(K)+B(K)*B(K))
С
С
                 COS(BETA(K)) = A(K)/ALPHA(K)
С
С
                  SIN(BETA(K)) = -B(K)/ALPHA(K)
С
**************
```

```
*****
С
     SUBROUTINE SINTI (N, WSAVE)
C
С
     SUBROUTINE SINTI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     SUBROUTINE SINT. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
     STORED IN WSAVE.
С
C
    INPUT PARAMETER
С
            THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N+1 IS A PRODUCT OF
SMALL PRIMES.
C
    OUTPUT PARAMETER
С
C
    WSAVE A WORK ARRAY WITH AT LEAST INT (2.5*N+15)
LOCATIONS.
             DIFFERENT WSAVE ARRAYS ARE REQUIRED FOR
DIFFERENT VALUES
             OF N. THE CONTENTS OF WSAVE MUST NOT BE
CHANGED BETWEEN
             CALLS OF SINT.
C
С
*************
*****
C
C
     SUBROUTINE SINT (N, X, WSAVE)
C
     SUBROUTINE SINT COMPUTES THE DISCRETE FOURIER SINE
TRANSFORM
     OF AN ODD SEQUENCE X(I). THE TRANSFORM IS DEFINED
BELOW AT
C
    OUTPUT PARAMETER X.
C.
    SINT IS THE UNNORMALIZED INVERSE OF ITSELF SINCE A
CALL OF SINT
C FOLLOWED BY ANOTHER CALL OF SINT WILL MULTIPLY THE
```

```
INPUT SEQUENCE
     X BY 2*(N+1).
C
C
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINT
MUST BE
     INITIALIZED BY CALLING SUBROUTINE SINTI (N, WSAVE).
C
С
     INPUT PARAMETERS
C
             THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N+1 IS THE PRODUCT
OF SMALL PRIMES.
C
С
     X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
TRANSFORMED
С
С
     WSAVE A WORK ARRAY WITH DIMENSION AT LEAST
INT(2.5*N+15)
              IN THE PROGRAM THAT CALLS SINT. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
SINTI(N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
              REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
C
              TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
С
С
  OUTPUT PARAMETERS
С
С
     X FOR I=1,\ldots,N
С
С
                  X(I) = THE SUM FROM K=1 TO K=N
С
С
                        2*X(K)*SIN(K*I*PI/(N+1))
С
С
                  A CALL OF SINT FOLLOWED BY ANOTHER
CALL OF
                  SINT WILL MULTIPLY THE SEQUENCE X BY
```

```
2*(N+1).
                HENCE SINT IS THE UNNORMALIZED
С
INVERSE
C
                OF ITSELF.
С
С
            CONTAINS INITIALIZATION CALCULATIONS WHICH
    WSAVE
MUST NOT BE
            DESTROYED BETWEEN CALLS OF SINT.
С
***************
******
С
     SUBROUTINE COSTI (N, WSAVE)
C
C
     SUBROUTINE COSTI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     SUBROUTINE COST. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
    STORED IN WSAVE.
C
    INPUT PARAMETER
С
C
    N THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED. THE METHOD
            IS MOST EFFICIENT WHEN N-1 IS A PRODUCT OF
SMALL PRIMES.
С
   OUTPUT PARAMETER
C
С
    WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
           DIFFERENT WSAVE ARRAYS ARE REQUIRED FOR
DIFFERENT VALUES
            OF N. THE CONTENTS OF WSAVE MUST NOT BE
CHANGED BETWEEN
            CALLS OF COST.
С
****************
*****
```

```
C SUBROUTINE COST (N, X, WSAVE)
С
C
      SUBROUTINE COST COMPUTES THE DISCRETE FOURIER
COSINE TRANSFORM
      OF AN EVEN SEQUENCE X(I). THE TRANSFORM IS DEFINED
BELOW AT OUTPUT
     PARAMETER X.
C
С
C
     COST IS THE UNNORMALIZED INVERSE OF ITSELF SINCE A
CALL OF COST
     FOLLOWED BY ANOTHER CALL OF COST WILL MULTIPLY THE
INPUT SEQUENCE
     X BY 2*(N-1). THE TRANSFORM IS DEFINED BELOW AT
OUTPUT PARAMETER X
C
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COST
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE COSTI (N, WSAVE).
С
С
     INPUT PARAMETERS
C
              THE LENGTH OF THE SEQUENCE X. N MUST BE
GREATER THAN 1.
              THE METHOD IS MOST EFFICIENT WHEN N-1 IS A
PRODUCT OF
С
             SMALL PRIMES.
С
     X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
C
TRANSFORMED
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15
              IN THE PROGRAM THAT CALLS COST. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
COSTI (N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
              VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
              REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSECUENT
              TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
```

```
С
С
     OUTPUT PARAMETERS
C
С
     Χ
            FOR I=1,\ldots,N
С
С
                 X(I) = X(1) + (-1) ** (I-1) *X(N)
С
С
                  + THE SUM FROM K=2 TO K=N-1
С
С
                      2*X(K)*COS((K-1)*(I-1)*PI/(N-1))
С
С
                  A CALL OF COST FOLLOWED BY ANOTHER
CALL OF
                  COST WILL MULTIPLY THE SEQUENCE X BY
2*(N-1)
С
                  HENCE COST IS THE UNNORMALIZED
INVERSE
                  OF ITSELF.
С
С
     WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT BE
             DESTROYED BETWEEN CALLS OF COST.
С
***************
*****
С
     SUBROUTINE SINQI (N, WSAVE)
С
С
     SUBROUTINE SINQI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     BOTH SINOF AND SINOB. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
     STORED IN WSAVE.
С
С
С
     INPUT PARAMETER
С
     N
             THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
```

```
C OUTPUT PARAMETER
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             THE SAME WORK ARRAY CAN BE USED FOR BOTH
SINQF AND SINQB
             AS LONG AS N REMAINS UNCHANGED. DIFFERENT
WSAVE ARRAYS
             ARE REQUIRED FOR DIFFERENT VALUES OF N.
THE CONTENTS OF
             WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF
SINQF OR SINQB.
C
****************
******
С
     SUBROUTINE SINOF (N, X, WSAVE)
С
     SUBROUTINE SINOF COMPUTES THE FAST FOURIER
TRANSFORM OF QUARTER
     WAVE DATA. THAT IS , SINQF COMPUTES THE
COEFFICIENTS IN A SINE
     SERIES REPRESENTATION WITH ONLY ODD WAVE NUMBERS.
THE TRANSFORM
     IS DEFINED BELOW AT OUTPUT PARAMETER X.
C
C
     SINQB IS THE UNNORMALIZED INVERSE OF SINQF SINCE A
CALL OF SINOF
     FOLLOWED BY A CALL OF SINQB WILL MULTIPLY THE
INPUT SEQUENCE X
     BY 4*N.
C
С
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINOF
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE SINQI (N, WSAVE).
С
С
С
     INPUT PARAMETERS
С
     N
             THE LENGTH OF THE ARRAY X TO BE
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
```

```
С
     X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
TRANSFORMED
C
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             IN THE PROGRAM THAT CALLS SINOF. THE WSAVE
ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
SINQI(N, WSAVE) AND A
             DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
             REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
C
С
     OUTPUT PARAMETERS
С
С
    X
            FOR I=1,\ldots,N
С
С
                  X(I) = (-1) ** (I-1) *X(N)
С
С
                     + THE SUM FROM K=1 TO K=N-1 OF
С
С
                     2*X(K)*SIN((2*I-1)*K*PI/(2*N))
С
С
                  A CALL OF SINQF FOLLOWED BY A CALL OF
С
                  SINOB WILL MULTIPLY THE SEQUENCE X BY
4*N.
                  THEREFORE SINOB IS THE UNNORMALIZED
С
INVERSE
                  OF SINOF.
C
С
     WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT
             BE DESTROYED BETWEEN CALLS OF SINOF OR
SINOB.
C
**************
******
```

```
С
С
     SUBROUTINE SINOB (N, X, WSAVE)
C
C
     SUBROUTINE SINOB COMPUTES THE FAST FOURIER
TRANSFORM OF QUARTER
     WAVE DATA. THAT IS , SINOB COMPUTES A SEQUENCE
FROM ITS
     REPRESENTATION IN TERMS OF A SINE SERIES WITH ODD
WAVE NUMBERS.
     THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER
Х.
C
C
      SINOF IS THE UNNORMALIZED INVERSE OF SINOB SINCE A
CALL OF SINOB
     FOLLOWED BY A CALL OF SINQF WILL MULTIPLY THE
INPUT SEQUENCE X
     BY 4*N.
С
С
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINOB
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE SINQI (N, WSAVE).
C
С
С
     INPUT PARAMETERS
С
              THE LENGTH OF THE ARRAY X TO BE
C
     Ν
TRANSFORMED. THE METHOD
              IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
С
    X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
TRANSFORMED
С
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15.
             IN THE PROGRAM THAT CALLS SINOB. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
SINQI (N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
         REPEATED SO LONG AS N REMAINS UNCHANGED
```

```
THUS SUBSEQUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
C
С
     OUTPUT PARAMETERS
С
С
    X FOR I=1,\ldots,N
С
С
                 X(I) = THE SUM FROM K=1 TO K=N OF
С
С
                    4*X(K)*SIN((2K-1)*I*PI/(2*N))
С
С
                 A CALL OF SINOB FOLLOWED BY A CALL OF
С
                  SINOF WILL MULTIPLY THE SEQUENCE X BY
4*N.
C
                 THEREFORE SINQF IS THE UNNORMALIZED
INVERSE
                 OF SINOB.
C
С
     WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT
            BE DESTROYED BETWEEN CALLS OF SINOB OR
SINOF.
С
****************
*****
С
    SUBROUTINE COSQI (N, WSAVE)
С
C.
     SUBROUTINE COSQI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     BOTH COSOF AND COSOB. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
С
    STORED IN WSAVE.
С
C INPUT PARAMETER
C
            THE LENGTH OF THE ARRAY TO BE TRANSFORMED.
    N
THE METHOD
             IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
```

```
С
    OUTPUT PARAMETER
C
     WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
C
LEAST 3*N+15.
             THE SAME WORK ARRAY CAN BE USED FOR BOTH
COSOF AND COSOB
             AS LONG AS N REMAINS UNCHANGED. DIFFERENT
WSAVE ARRAYS
             ARE REQUIRED FOR DIFFERENT VALUES OF N.
THE CONTENTS OF
             WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF
COSOF OR COSOB.
C
С
*****************
*****
С
     SUBROUTINE COSQF (N, X, WSAVE)
С
C
     SUBROUTINE COSOF COMPUTES THE FAST FOURIER
TRANSFORM OF QUARTER
     WAVE DATA. THAT IS , COSQF COMPUTES THE
COEFFICIENTS IN A COSINE
     SERIES REPRESENTATION WITH ONLY ODD WAVE NUMBERS.
THE TRANSFORM
     IS DEFINED BELOW AT OUTPUT PARAMETER X
C
     COSOF IS THE UNNORMALIZED INVERSE OF COSOB SINCE A
CALL OF COSQF
     FOLLOWED BY A CALL OF COSQB WILL MULTIPLY THE
INPUT SEOUENCE X
    BY 4*N.
С
C
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COSOF
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE COSQI (N, WSAVE).
С
С
C
     INPUT PARAMETERS
C
С
            THE LENGTH OF THE ARRAY X TO BE
     Ν
TRANSFORMED. THE METHOD
          IS MOST EFFICIENT WHEN N IS A PRODUCT OF
```

```
SMALL PRIMES.
C
     X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
TRANSFORMED
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
LEAST 3*N+15
            IN THE PROGRAM THAT CALLS COSOF. THE WSAVE
ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
COSQI (N, WSAVE) AND A
            DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
            VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
            REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSECUENT
            TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
C OUTPUT PARAMETERS
C
С
    X FOR I=1,\ldots,N
С
С
                 X(I) = X(1) PLUS THE SUM FROM K=2 TO
K=N OF
C
С
                    2*X(K)*COS((2*I-1)*(K-1)*PI/(2*N))
С
С
                 A CALL OF COSQF FOLLOWED BY A CALL OF
C
                 COSQB WILL MULTIPLY THE SEQUENCE X BY
4*N.
                 THEREFORE COSOB IS THE UNNORMALIZED
С
INVERSE
                 OF COSOF.
C
C
    WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT
С
            BE DESTROYED BETWEEN CALLS OF COSOF OR
COSQB.
*************
*****
```

```
С
С
     SUBROUTINE COSQB (N, X, WSAVE)
C
     SUBROUTINE COSOB COMPUTES THE FAST FOURIER
TRANSFORM OF QUARTER
     WAVE DATA. THAT IS , COSQB COMPUTES A SEQUENCE
FROM ITS
     REPRESENTATION IN TERMS OF A COSINE SERIES WITH
ODD WAVE NUMBERS.
     THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER
Χ.
C
C
     COSOB IS THE UNNORMALIZED INVERSE OF COSOF SINCE A
CALL OF COSOB
     FOLLOWED BY A CALL OF COSQF WILL MULTIPLY THE
INPUT SEQUENCE X
     BY 4*N.
С
С
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COSOB
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE COSQI (N, WSAVE).
C
С
С
     INPUT PARAMETERS
С
             THE LENGTH OF THE ARRAY X TO BE
C
     Ν
TRANSFORMED. THE METHOD
             IS MOST EFFICIENT WHEN N IS A PRODUCT OF
SMALL PRIMES.
С
    X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE
TRANSFORMED
C
     WSAVE A WORK ARRAY THAT MUST BE DIMENSIONED AT
LEAST 3*N+15
             IN THE PROGRAM THAT CALLS COSOB. THE WSAVE
ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
COSQI (N, WSAVE) AND A
             DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
         REPEATED SO LONG AS N REMAINS UNCHANGED
```

```
THUS SUBSECUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
C
С
     OUTPUT PARAMETERS
С
С
    X FOR I=1,\ldots,N
С
С
                 X(I) = THE SUM FROM K=1 TO K=N OF
С
С
                   4*X(K)*COS((2*K-1)*(I-1)*PI/(2*N))
С
С
                 A CALL OF COSOB FOLLOWED BY A CALL OF
С
                 COSOF WILL MULTIPLY THE SEQUENCE X BY
4*N.
C
                 THEREFORE COSOF IS THE UNNORMALIZED
INVERSE
                 OF COSOB.
C
С
    WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT
            BE DESTROYED BETWEEN CALLS OF COSOB OR
COSOF.
С
****************
*****
С
    SUBROUTINE CFFTI (N, WSAVE)
С
     SUBROUTINE CFFTI INITIALIZES THE ARRAY WSAVE WHICH
IS USED IN
     BOTH CFFTF AND CFFTB. THE PRIME FACTORIZATION OF N
TOGETHER WITH
     A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE
COMPUTED AND
С
    STORED IN WSAVE.
С
C INPUT PARAMETER
C
    N THE LENGTH OF THE SEQUENCE TO BE
TRANSFORMED
С
C OUTPUT PARAMETER
```

```
С
    WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT
С
LEAST 4*N+15
            THE SAME WORK ARRAY CAN BE USED FOR BOTH
CFFTF AND CFFTB
             AS LONG AS N REMAINS UNCHANGED. DIFFERENT
WSAVE ARRAYS
            ARE REQUIRED FOR DIFFERENT VALUES OF N.
THE CONTENTS OF
             WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF
CFFTF OR CFFTB.
**************
******
С
     SUBROUTINE CFFTF (N, C, WSAVE)
С
     SUBROUTINE CFFTF COMPUTES THE FORWARD COMPLEX
DISCRETE FOURIER
     TRANSFORM (THE FOURIER ANALYSIS). EQUIVALENTLY ,
CFFTF COMPUTES
     THE FOURIER COEFFICIENTS OF A COMPLEX PERIODIC
SEQUENCE.
     THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER
С.
C
     THE TRANSFORM IS NOT NORMALIZED. TO OBTAIN A
NORMALIZED TRANSFORM
     THE OUTPUT MUST BE DIVIDED BY N. OTHERWISE A CALL
OF CFFTF
     FOLLOWED BY A CALL OF CFFTB WILL MULTIPLY THE
SEQUENCE BY N.
C
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE CFFTF
MUST BE
С
     INITIALIZED BY CALLING SUBROUTINE CFFTI (N, WSAVE).
C
С
    INPUT PARAMETERS
С
C
C
    N
           THE LENGTH OF THE COMPLEX SEQUENCE C. THE
METHOD IS
      MORE EFFICIENT WHEN N IS THE PRODUCT OF
```

```
SMALL PRIMES. N
C
    C A COMPLEX ARRAY OF LENGTH N WHICH CONTAINS
THE SEQUENCE
     WSAVE A REAL WORK ARRAY WHICH MUST BE
DIMENSIONED AT LEAST 4N+15
             IN THE PROGRAM THAT CALLS CFFTF. THE WSAVE
ARRAY MUST BE
             INITIALIZED BY CALLING SUBROUTINE
CFFTI (N, WSAVE) AND A
            DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
            VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
            REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
            TRANSFORMS CAN BE OBTAINED FASTER THAN THE
FIRST.
            THE SAME WSAVE ARRAY CAN BE USED BY CFFTF
AND CFFTB.
    OUTPUT PARAMETERS
С
С
С
    C FOR J=1,\ldots,N
С
С
                C(J) = THE SUM FROM K=1,...,N OF
С
С
                     C(K) *EXP(-I*(J-1)*(K-1)*2*PI/N)
С
С
                           WHERE I=SQRT(-1)
C
С
     WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH
MUST NOT BE
            DESTROYED BETWEEN CALLS OF SUBROUTINE
CFFTF OR CFFTB
С
*************
*****
С
     SUBROUTINE CFFTB (N, C, WSAVE)
C
C SUBROUTINE CFFTB COMPUTES THE BACKWARD COMPLEX
```

```
DISCRETE FOURIER
     TRANSFORM (THE FOURIER SYNTHESIS). EQUIVALENTLY,
CFFTB COMPUTES
     A COMPLEX PERIODIC SEQUENCE FROM ITS FOURIER
COEFFICIENTS.
     THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER
C
С.
С
     A CALL OF CFFTF FOLLOWED BY A CALL OF CFFTB WILL
MULTIPLY THE
С
     SEQUENCE BY N.
С
C
     THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE CFFTB
MUST BE
     INITIALIZED BY CALLING SUBROUTINE CFFTI (N, WSAVE).
С
С
     INPUT PARAMETERS
С
C
С
     N THE LENGTH OF THE COMPLEX SEQUENCE C. THE
METHOD IS
           MORE EFFICIENT WHEN N IS THE PRODUCT OF
SMALL PRIMES.
С
    C A COMPLEX ARRAY OF LENGTH N WHICH CONTAINS
THE SEOUENCE
C
С
     WSAVE A REAL WORK ARRAY WHICH MUST BE
DIMENSIONED AT LEAST 4N+15
             IN THE PROGRAM THAT CALLS CFFTB. THE WSAVE
ARRAY MUST BE
              INITIALIZED BY CALLING SUBROUTINE
CFFTI (N, WSAVE) AND A
              DIFFERENT WSAVE ARRAY MUST BE USED FOR
EACH DIFFERENT
             VALUE OF N. THIS INITIALIZATION DOES NOT
HAVE TO BE
             REPEATED SO LONG AS N REMAINS UNCHANGED
THUS SUBSEQUENT
             TRANSFORMS CAN BE OBTAINED FASTER THAN THE
C
FIRST.
С
             THE SAME WSAVE ARRAY CAN BE USED BY CFFTF
AND CFFTB.
```

```
OUTPUT PARAMETERS
С
С
         FOR J=1,\ldots,N
С
С
              C(J) = THE SUM FROM K=1,...,N OF
С
                    C(K) *EXP(I*(J-1)*(K-1)*2*PI/N)
С
С
С
                         WHERE I=SQRT(-1)
С
С
            CONTAINS INITIALIZATION CALCULATIONS WHICH
     WSAVE
MUST NOT BE
            DESTROYED BETWEEN CALLS OF SUBROUTINE
CFFTF OR CFFTB
*************
*****
```

GENBUN

```
FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
С
                     the National Science Foundation
```

```
C
С
      SUBROUTINE GENBUN
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y, IERROR)
С
C DIMENSION OF
                          A(M), B(M), C(M), Y(IDIMY, N)
C ARGUMENTS
C LATEST REVISION
                         JUNE 2004
C PURPOSE
                          THE NAME OF THIS PACKAGE IS A
MNEMONIC FOR THE
                          GENERALIZED BUNEMAN ALGORITHM.
С
C
                           IT SOLVES THE REAL LINEAR
SYSTEM OF EQUATIONS
С
C
                          A(I) * X(I-1,J) + B(I) * X(I,J) +
C(I) *X(I+1,J)
                          + X(I,J-1) - 2.*X(I,J) +
X(I,J+1) = Y(I,J)
C
                          FOR I = 1, 2, \ldots, M AND J =
1, 2, ..., N.
                           INDICES I+1 AND I-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 EQUAL 0, X(I,2),
OR X(I,N),
                          AND X(I,N+1) MAY EQUAL 0,
C
X(I,N-1), OR X(I,1)
                          DEPENDING ON AN INPUT
PARAMETER.
C
C USAGE
                           CALL GENBUN
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y,
С
                                         IERROR)
С
```

```
C ARGUMENTS
C
C ON INPUT
                            NPEROD
C
С
                               INDICATES THE VALUES THAT
X(I,0) AND
                               X(I,N+1) ARE ASSUMED TO HAVE.
C
С
С
                               = 0 	ext{ IF } X(I,0) = X(I,N) 	ext{ AND}
X(I,N+1) =
                                     X(I,1).
С
                               = 1 	ext{ IF } X(I,0) = X(I,N+1) = 0
                               = 2 \quad \text{IF } X(I,0) = 0 \quad \text{AND}
X(I,N+1) = X(I,N-1).
                               = 3 	ext{ IF } X(I,0) = X(I,2) 	ext{ AND}
X(I,N+1) =
                                     X(I, N-1).
C
C
                               = 4 \quad \text{IF } X(I,0) = X(I,2) \quad \text{AND}
X(I,N+1) = 0.
С
С
                            N
C
                               THE NUMBER OF UNKNOWNS IN THE
J-DIRECTION.
                               N MUST BE GREATER THAN 2.
С
С
                             MPEROD
С
                               = 0 IF A(1) AND C(M) ARE NOT
ZERO
С
                               = 1 \text{ IF A (1)} = C(M) = 0
С
С
                             M
С
                               THE NUMBER OF UNKNOWNS IN THE
I-DIRECTION.
                               N MUST BE GREATER THAN 2.
С
С
С
                            A,B,C
С
                               ONE-DIMENSIONAL ARRAYS OF
LENGTH M THAT
                               SPECIFY THE COEFFICIENTS IN
THE LINEAR
                               EOUATIONS GIVEN ABOVE. IF
MPEROD = 0
                               THE ARRAY ELEMENTS MUST NOT
```

DEPEND UPON	
C	THE INDEX I, BUT MUST BE
CONSTANT.	
C	SPECIFICALLY, THE SUBROUTINE
CHECKS THE	
C	FOLLOWING CONDITION .
C	
С	A(I) = C(1)
С	C(I) = C(1)
С	B(I) = B(1)
С	
C	FOR $I=1,2,,M$.
C	
C	IDIMY
C	THE ROW (OR FIRST) DIMENSION
OF THE	
C	TWO-DIMENSIONAL ARRAY Y AS IT
APPEARS	
C	IN THE PROGRAM CALLING
GENBUN.	
C	THIS PARAMETER IS USED TO
SPECIFY THE	
C	VARIABLE DIMENSION OF Y.
C	IDIMY MUST BE AT LEAST M.
C	
С	Y
C	A TWO-DIMENSIONAL COMPLEX
ARRAY THAT	
C	SPECIFIES THE VALUES OF THE
RIGHT SIDE	
C	OF THE LINEAR SYSTEM OF
EQUATIONS GIVEN	
C	ABOVE.
C	Y MUST BE DIMENSIONED AT
LEAST M*N.	
C	
C	
C ON OUTPUT	Y
С	
С	CONTAINS THE SOLUTION X.
С	
С	IERROR
С	AN ERROR FLAG WHICH INDICATES
INVALID	

C	INPUT PARAMETERS EXCEPT FOR
NUMBER	CEDO A COLUMNON TO NOM
C	ZERO, A SOLUTION IS NOT
ATTEMPTED.	
C	0 110 EDDOD
C	= 0 NO ERROR.
С	= 1 M .LE. 2 .
C	= 2 N .LE. 2
C	= 3 IDIMY .LT. M
C	= 4 NPEROD .LT. 0 OR NPEROD
.GT. 4	
C	= 5 MPEROD .LT. 0 OR MPEROD
.GT. 1	
C	= 6 A(I) .NE. C(I) OR C(I)
.NE. C(1) OR	
C C	B(I) .NE. $B(1)$ FOR
C	SOME $I=1,2,\ldots,M$.
C	= 7 A(1) . NE. 0 OR C(M) . NE.
0 AND	7 11(1) 0 01(0 (11)
C	MPEROD = 1
C	= 20 If the dynamic
allocation of real and	
C	complex work space
required for solution	
C	fails (for example if
N,M are too large	
C	for your computer)
C	
C	
C SPECIAL CONDITONS	NONE
C	
C I/O	NONE
C	
C PRECISION	SINGLE
C	
C REQUIRED FILES	comf.f,gnbnaux.f,fish.f
C FILES	comi.i, gimiaax.i, iioii.i
C	
C LANGUAGE	FORTRAN 90
	LOITIVAIN 30
С	MOTOMENI IN 1070 DV DOLAND COMPA
C HISTORY	WRITTEN IN 1979 BY ROLAND SWEET
OF NCAR'S	
C	SCIENTIFIC COMPUTING DIVISION.
MADE AVAILABLE	

C	ON NCAR'S PUBLIC LIBRARIES IN
JANUARY, 1980.	
С	Revised in June 2004 by John
Adams using	
С	Fortran 90 dynamically
allocated work space.	
С	
C ALGORITHM	THE LINEAR SYSTEM IS SOLVED BY
A CYCLIC	
C	REDUCTION ALGORITHM DESCRIBED
IN THE	DEFEDENCE
C	REFERENCE.
C PORTABILITY	FORTRAN 90
C	THE MACHINE DEPENDENT CONSTANT
PI IS	THE TRICITIVE DELENDED CONDITANT
C	DEFINED IN FUNCTION PIMACH.
C	
C REFERENCES	SWEET, R., "A CYCLIC REDUCTION
ALGORITHM FOR	
С	SOLVING BLOCK TRIDIAGONAL
SYSTEMS OF ARBITRARY	
С	DIMENSIONS," SIAM J. ON NUMER.
ANAL., 14 (1977)	506 500
C	PP. 706-720.
C	MILLO MECH MAC DEDECORMED ON O
C ACCURACY platform with	THIS TEST WAS PERFORMED ON a
C PIACIOIII WICH	64 bit floating point
arithmetic.	of Die Hoating point
C	A UNIFORM RANDOM NUMBER
GENERATOR WAS USED	01 01 1 01- 1-01
C	TO CREATE A SOLUTION ARRAY X
FOR THE SYSTEM	
С	GIVEN IN THE 'PURPOSE'
DESCRIPTION ABOVE	
С	WITH
С	A(I) = C(I) = -0.5*B(I) = 1,
I=1,2,,M	
C	1110 MININ MPDD 02 1
C	AND, WHEN MPEROD = 1
C	7(1) - C(M) - 0
C	A(1) = C(M) = 0 A(M) = C(1) = 2.
	A(P) - C(1) - 2.

C	THI	E SOLUTION	X WAS SUBSI	TITUTED
INTO THE				
C PRECISION	GI	VEN SYSTEM	AND, USING	DOUBLE
C			Y WAS COMPU	
C GENBUN	US:	ING THIS A	RRAY Y, SUBF	ROUTINE
C	WAS	S CALLED TO) PRODUCE	
APPROXIMATE	CO.	TIMTON 7	חוודאו הדד אחיו	- T 7T-7
C ERROR	50.	LUIION Z.	THEN RELATI	- VĽ
C	Ι		S(Z(I,J)-X(I	(, J)))/
C	WAS	•	S(X(I,J))) , WHERE THE	TWO
MAXIMA ARE TAKEN		1 0	4	
C	OVI	ER $I=1,2,$,M AND J=1	.,,N.
C	THI	E VALUE OF	E IS GIVEN	IN THE
TABLE C	BEI	OW FOR SOM	Æ TYPICAL V	/ATJUES
OF M AND N.				
C	M (=N)	MPEROD	NPEROD	E
C				
 C				
C	31	0	0	6.E-
14 C	31	1	1	4.E-
13)I	Τ.	1	4.17
C	31	1	3	3.E-
13 C	32	0	0	9.E-
14	22	1	1	2 12
C 13	32	1	1	3.E-
C	32	1	3	1.E-
13 C	33	0	0	9.E-
14				
C 13	33	1	1	4.E-
C	33	1	3	1.E-

13				
C	63	0	0	1.E-
13	62	1	1	1 🖽
C 12	63	1	1	1.E-
C	63	1	3	2.E-
13				
C	64	0	0	1.E-
13 C	64	1	1	1.E-
12	04	_	_	T • E
C	64	1	3	6.E-
13				_
C 13	65	0	0	2.E-
C	65	1	1	1.E-
12		_	_	
C	65	1	3	4.E-
13	le ale ale ale al 1	de de de d	de de de d	
* * * * * * * * * * * * * * * * * * *	* * * * *	* * * * *	* * * * *	* *

GNBNAUX

```
all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
```

C *	
*	
	e National Science Foundation
*	
C *	
*	
* * * * * * * * * * * * * * * * * * *	* * * * * * * * * * * * * * * * * *
C DACKACE CNIDNIALIN	
C PACKAGE GNBNAUX	
C LATEST REVISION	June 2004
C LATEST REVISION	ourie 2004
C PURPOSE	TO PROVIDE AUXILIARY ROUTINES
FOR FISHPAK	
C	ENTRIES GENBUN AND POISTG.
C	
C USAGE	THERE ARE NO USER ENTRIES IN
THIS PACKAGE.	
C	THE ROUTINES IN THIS PACKAGE
ARE NOT INTENDED	
C	TO BE CALLED BY USERS, BUT
RATHER BY ROUTINES	
C	IN PACKAGES GENBUN AND POISTG.
C C C C C C C C C C C C C C C C C C C	NOVE
C SPECIAL CONDITIONS	NONE
C I/O	NONE
C 1/0	NONE
C PRECISION	SINGLE
C	21.922
C REQUIRED files	fish.f, comf.f
C	·
C LANGUAGE	FORTRAN 90
C	
C HISTORY	WRITTEN IN 1979 BY ROLAND SWEET
OF NCAR'S	
С	SCIENTIFIC COMPUTING DIVISION.
MADE AVAILABLE	
C TANTA DAY 1000	ON NCAR'S PUBLIC LIBRARIES IN
JANUARY, 1980.	Desired in Tax 2004 1 Th
C Adama ugina	Revised in June 2004 by John
Adams using	Fortron 00 dimenically
C	Fortran 90 dynamically

HSTCRT

```
file hstcrt.txt (documentation for the FISHPACK
solver HSTCRT)
* * * * * * *
С
                         copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
С
                            all rights reserved
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
*
С
                       Subroutines and Example Programs
```

```
С
С
                        for Modeling Geophysical Processes
С
С
                                       by
С
С
                John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                       of
С
С
                 the National Center for Atmospheric
Research
                         Boulder, Colorado (80307)
U.S.A.
С
                            which is sponsored by
С
*
С
С
                      the National Science Foundation
С
С
С
      SUBROUTINE HSTCRT
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                           ELMBDA, F, IDIMF, PERTRB, IERROR)
С
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N)
```

```
C ARGUMENTS
С
C LATEST REVISION June 2004
C PURPOSE
                         SOLVES THE STANDARD FIVE-POINT
FINITE
                          DIFFERENCE APPROXIMATION TO
THE HELMHOLTZ
                          EQUATION
                            (D/DX)(DU/DX) +
(D/DY)(DU/DY) + LAMBDA*U
                           = F(X,Y)
                          ON A STAGGERED GRID IN
CARTESIAN COORDINATES.
C USAGE
                          CALL HSTCRT
(A,B,M,MBDCND,BDA,BDB,C,D
N, NBDCND, BDC, BDD, ELMBDA,
F, IDIMF, PERTRB, IERROR)
C ARGUMENTS
C ON INPUT
С
С
                         A,B
C
                           THE RANGE OF X, I.E. A .LE. X
.LE. B.
                           A MUST BE LESS THAN B.
C
С
С
                           THE NUMBER OF GRID POINTS IN
THE
                           INTERVAL (A, B). THE GRID
POINTS
С
                           IN THE X-DIRECTION ARE GIVEN
BY
С
                           X(I) = A + (I-0.5)DX FOR
I=1,2,...,M
                           WHERE DX = (B-A)/M. M MUST BE
GREATER
                           THAN 2.
С
С
                         MBDCND
```

С	INDICATES THE TYPE OF
BOUNDARY CONDITIONS	
C	AT X = A AND X = B.
C	
C	= 0 IF THE SOLUTION IS
PERIODIC IN X,	
С	U(M+I,J) = U(I,J).
С	
C	= 1 IF THE SOLUTION IS
SPECIFIED AT	
C	X = A AND X = B.
С	•
С	= 2 IF THE SOLUTION IS
SPECIFIED AT	
C	X = A AND THE DERIVATIVE
C	OF THE SOLUTION WITH
RESPECT TO X	
C	IS SPECIFIED AT $X = B$.
С	_
C	= 3 IF THE DERIVATIVE OF THE
SOLUTION	
C	WITH RESPECT TO X IS
SPECIFIED	
C	AT $X = A$ AND $X = B$.
C	4
C	= 4 IF THE DERIVATIVE OF THE
SOLUTION	LITTU DEGDECE EO U TO
C	WITH RESPECT TO X IS
SPECIFIED	
C	AT X = A AND THE
SOLUTION IS	
C	SPECIFIED AT $X = B$.
C	DD3
C	BDA
	A ONE-DIMENSIONAL ARRAY OF
LENGTH N	MINE COECTETES MIE DOINDADY
C	THAT SPECIFIES THE BOUNDARY
VALUES	(TE ANV) OF MITE COTIMITON AM V
C _ 7	(IF ANY) OF THE SOLUTION AT X
= A.	
C	WHEN MODOND - 1 OD 2
C	WHEN MBDCND = 1 OR 2, $RDA(T) = II(A, Y(T))$
C N	BDA(J) = U(A, Y(J)) ,
J=1,2,,N.	

```
С
                            WHEN MBDCND = 3 \text{ OR } 4,
                              BDA(J) = (D/DX)U(A,Y(J)),
C
J=1,2,...,N.
С
                          BDB
C
                            A ONE-DIMENSIONAL ARRAY OF
LENGTH N
                            THAT SPECIFIES THE BOUNDARY
VALUES
                            OF THE SOLUTION AT X = B.
С
С
С
                            WHEN MBDCND = 1 OR 4
С
                              BDB(J) = U(B,Y(J)),
J=1,2,...,N.
С
С
                            WHEN MBDCND = 2 OR 3
С
                              BDB(J) = (D/DX)U(B,Y(J)),
J=1,2,...,N.
С
                          C,D
C
                            THE RANGE OF Y, I.E. C .LE. Y
.LE. D.
                            C MUST BE LESS THAN D.
С
С
С
С
                            THE NUMBER OF UNKNOWNS IN THE
INTERVAL
                            (C,D). THE UNKNOWNS IN THE
Y-DIRECTION
                            ARE GIVEN BY Y(J) = C + (J -
0.5) DY,
                            J=1,2,\ldots,N, WHERE DY = (D-
C)/N.
С
                            N MUST BE GREATER THAN 2.
С
С
                          NBDCND
                            INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                            AT Y = C AND Y = D.
С
\mathsf{C}
                            = 0 IF THE SOLUTION IS
```

```
PERIODIC IN Y, I.E.
С
                                U(I,J) = U(I,N+J).
C
                           = 1 IF THE SOLUTION IS
C
SPECIFIED AT Y = C
                                AND Y = D.
С
С
                           = 2 IF THE SOLUTION IS
SPECIFIED AT Y = C
                                AND THE DERIVATIVE OF
THE SOLUTION
                                WITH RESPECT TO Y IS
SPECIFIED AT
                                Y = D.
С
С
                           = 3 IF THE DERIVATIVE OF THE
SOLUTION
                                WITH RESPECT TO Y IS
SPECIFIED AT
                                Y = C AND Y = D.
С
                           = 4 IF THE DERIVATIVE OF THE
SOLUTION
                                WITH RESPECT TO Y IS
SPECIFIED AT
                                Y = C AND THE SOLUTION
C
IS SPECIFIED
                                AT Y = D.
С
С
С
                         BDC
                           A ONE DIMENSIONAL ARRAY OF
LENGTH M THAT
                           SPECIFIES THE BOUNDARY VALUES
C
OF THE
С
                           SOLUTION AT Y = C.
С
С
                           WHEN NBDCND = 1 \text{ OR } 2,
                             BDC(I) = U(X(I),C),
С
I=1,2,...,M.
C
C
                           WHEN NBDCND = 3 OR 4,
С
                             BDC(I) = (D/DY)U(X(I),C),
I=1,2,...,M.
```

```
WHEN NBDCND = 0, BDC IS A
DUMMY VARIABLE.
С
С
                         BDD
С
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH M THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
С
                            SOLUTION AT Y = D.
С
С
                            WHEN NBDCND = 1 OR 4,
С
                             BDD(I) = U(X(I),D),
I=1,2,...,M.
C
С
                            WHEN NBDCND = 2 \text{ OR } 3,
                              BDD(I) = (D/DY)U(X(I),D) ,
С
I=1,2,...,M.
C
                            WHEN NBDCND = 0, BDD IS A
DUMMY VARIABLE.
С
                         ELMBDA
C
                            THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                           EQUATION. IF LAMBDA IS
GREATER THAN 0,
                           A SOLUTION MAY NOT EXIST.
HOWEVER,
                           HSTCRT WILL ATTEMPT TO FIND
A SOLUTION.
С
С
                         F
С
                           A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE RIGHT SIDE
OF THE
                           HELMHOLTZ EQUATION. FOR
I=1,2,...,M
С
                           AND J=1,2,\ldots,N
С
С
                             F(I,J) = F(X(I),Y(J)) .
С
С
                           F MUST BE DIMENSIONED AT
LEAST M X N.
```

C	
C	IDIMF
C	THE ROW (OR FIRST) DIMENSION
OF THE ARRAY	
С	F AS IT APPEARS IN THE
PROGRAM CALLING	
C	HSTCRT. THIS PARAMETER IS
USED TO SPECIFY	
C	THE VARIABLE DIMENSION OF F.
C	IDIMF MUST BE AT LEAST M.
C	IDIII 11001 DD 711 DD 101 11.
C	
	F
	-
C	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	(X(I),Y(J)) FOR $I=1,2,,M$,
J=1,2,,N.	
C	
C	PERTRB
C	IF A COMBINATION OF PERIODIC
OR DERIVATIVE	
C	BOUNDARY CONDITIONS IS
SPECIFIED FOR A	
C	DOTCCON EQUATION (TAMBDA -
	POISSON FOUATION CLAMBDA -
	POISSON EQUATION (LAMBDA =
0), A SOLUTION	
0), A SOLUTION C	MAY NOT EXIST. PERTRB IS A
0), A SOLUTION C CONSTANT,	MAY NOT EXIST. PERTRB IS A
0), A SOLUTION C CONSTANT, C	
0), A SOLUTION C CONSTANT, C FROM F, WHICH	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED
0), A SOLUTION C CONSTANT, C FROM F, WHICH C	MAY NOT EXIST. PERTRB IS A
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION,
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION,
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION,
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE
O), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE
O), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL C PLUS ANY C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE APPROXIMATION. THIS SOLUTION
0), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL C PLUS ANY	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE APPROXIMATION. THIS SOLUTION CONSTANT IS ALSO A SOLUTION;
O), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL C PLUS ANY C HENCE, THE C	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE APPROXIMATION. THIS SOLUTION
O), A SOLUTION C CONSTANT, C FROM F, WHICH C EXISTS. HSTCRT C WHICH IS A C ORIGINAL C PLUS ANY C HENCE, THE	MAY NOT EXIST. PERTRB IS A CALCULATED AND SUBTRACTED ENSURES THAT A SOLUTION THEN COMPUTES THIS SOLUTION, LEAST SQUARES SOLUTION TO THE APPROXIMATION. THIS SOLUTION CONSTANT IS ALSO A SOLUTION;

COMPARED TO THE	
C	RIGHT SIDE F. OTHERWISE, A
SOLUTION IS	
C	OBTAINED TO AN ESSENTIALLY
DIFFERENT PROBLEM.	
C	THIS COMPARISON SHOULD ALWAYS
BE MADE TO	THOUSE THE RESIDENCE OF
C COLUMN IN C. DEEN	INSURE THAT A MEANINGFUL
SOLUTION HAS BEEN	ODMATNED
C	OBTAINED.
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	AN ERROR PLAG HIAI INDICATES
C C	PARAMETERS. EXCEPT TO
NUMBERS 0 AND 6,	TAIVAMETERO: EXCELT TO
C	A SOLUTION IS NOT ATTEMPTED.
C	
C	= 0 NO ERROR
С	
С	= 1 A .GE. B
C	
C	= 2 MBDCND .LT. 0 OR MBDCND
.GT. 4	
С	
C	= 3 C .GE. D
C	4 0
C	= 4 N .LE. 2
C	- E NIDDOND IE O OD NIDDOND
.GT. 4	= 5 NBDCND .LT. 0 OR NBDCND
.G1. 4	
C	= 6 LAMBDA .GT. 0
C	
C	= 7 IDIMF .LT. M
C	,
C	= 8 M .LE. 2
С	
C	SINCE THIS IS THE ONLY MEANS
OF INDICATING	
С	A POSSIBLY INCORRECT CALL TO
HSTCRT, THE	
С	USER SHOULD TEST IERROR AFTER
THE CALL.	

```
С
                          = 20 If the dynamic
allocation of real and
                                complex work space
required for solution
                                fails (for example if
N,M are too large
                                for your computer)
С
С
C I/O
                        NONE
C PRECISION
                        SINGLE
C REQUIRED LIBRARY
fish.f, comf.f, genbun.f, gnbnaux.f, poistg.f
C FILES
С
C LANGUAGE
                FORTRAN 90
C HISTORY
                       WRITTEN BY ROLAND SWEET AT NCAR
IN 1977.
                        RELEASED ON NCAR'S PUBLIC
SOFTWARE LIBRARIES
                         IN JANUARY 1980.
                        Revised in June 2004 by John
Adams using
                        Fortran 90 dynamically
allocated work space.
C PORTABILITY
                        FORTRAN 90
C ALGORITHM
                         THIS SUBROUTINE DEFINES THE
FINITE-DIFFERENCE
                         EQUATIONS, INCORPORATES
BOUNDARY DATA, ADJUSTS
                         THE RIGHT SIDE WHEN THE SYSTEM
IS SINGULAR
                         AND CALLS EITHER POISTG OR
GENBUN WHICH SOLVES
                         THE LINEAR SYSTEM OF EQUATIONS.
С
C TIMING
                         FOR LARGE M AND N, THE
OPERATION COUNT
```

IS ROUGHLY PROPORTIONAL TO M*N*LOG2(N). С C ACCURACY THE SOLUTION PROCESS EMPLOYED RESULTS IN A LOSS OF NO MORE THAN FOUR SIGNIFICANT DIGITS FOR N AND M AS LARGE AS 64. MORE DETAILED INFORMATION ABOUT ACCURACY CAN BE FOUND IN THE DOCUMENTATION FOR PACKAGE POISTG WHICH SOLVES THE FINITE DIFFERENCE EQUATIONS. C REFERENCES U. SCHUMANN AND R. SWEET, "A DIRECT METHOD FOR THE SOLUTION OF POISSON'S EQUATION WITH BOUNDARY CONDITIONS ON A STAGGERED GRID OF ARBITRARY SIZE," J. COMP. PHYS. 20 (1976), PP. 171-182. *****

HSTCSP

```
copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
С
Research
                              all rights reserved
С
С
С
                            FISHPACK90 version 1.1
С
                         A Package of Fortran 77 and 90
С
С
                        Subroutines and Example Programs
С
С
                       for Modeling Geophysical Processes
С
С
С
                                      by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                      of
С
                the National Center for Atmospheric
Research
\mathsf{C}
```

```
C *
                        Boulder, Colorado (80307)
U.S.A.
C
С
                           which is sponsored by
C
С
                      the National Science Foundation
С
C
С
      SUBROUTINE HSTCSP
(INTL, A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC,
BDD, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
С
C
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N)
C ARGUMENTS
C
C LATEST REVISION
                   June 2004
C PURPOSE
                          SOLVES THE STANDARD FIVE-POINT
FINITE
                          DIFFERENCE APPROXIMATION ON A
STAGGERED
                          GRID TO THE MODIFIED HELMHOLTZ
EQUATION IN
                          SPHERICAL COORDINATES ASSUMING
AXISYMMETRY
С
                          (NO DEPENDENCE ON LONGITUDE).
С
С
                          THE EQUATION IS
С
С
                              (1/R**2) (D/DR) (R**2 (DU/DR))
+
С
1/(R**2*SIN(THETA))(D/DTHETA)
                              (SIN(THETA)(DU/DTHETA)) +
```

```
(LAMBDA/(R*SIN(THETA))**2)U
= F (THETA, R)
С
                         WHERE THETA IS COLATITUDE AND R
C
IS THE
                         RADIAL COORDINATE. THIS TWO-
DIMENSIONAL
                         MODIFIED HELMHOLTZ EQUATION
RESULTS FROM
                         THE FOURIER TRANSFORM OF THE
THREE-
C
                         DIMENSIONAL POISSON EQUATION.
С
С
C USAGE
                         CALL HSTCSP
(INTL, A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
C
                                       PERTRB, IERROR, W)
С
C ARGUMENTS
C ON INPUT
                        INTL
С
                           = 0 ON INITIAL ENTRY TO
HSTCSP OR IF ANY
C
                                OF THE ARGUMENTS C, D,
N, OR NBDCND
                                ARE CHANGED FROM A
PREVIOUS CALL
С
C
                           = 1 IF C, D, N, AND NBDCND
ARE ALL
                                UNCHANGED FROM PREVIOUS
CALL TO HSTCSP
С
                           NOTE:
                           A CALL WITH INTL = 0 TAKES
APPROXIMATELY
                           1.5 TIMES AS MUCH TIME AS A
CALL WITH
                           INTL = 1. ONCE A CALL WITH
INTL = 0
                           HAS BEEN MADE THEN SUBSEQUENT
SOLUTIONS
```

C	CORRESPONDING TO DIFFERENT F,
BDA, BDB,	BDC, AND BDD CAN BE OBTAINED
FASTER WITH	DDO, TEND DDD OF THE CONTINUED
C	INTL = 1 SINCE INITIALIZATION
IS NOT	
С	REPEATED.
С	
С	A, B
C	THE RANGE OF THETA
(COLATITUDE),	
С	I.E. A .LE. THETA .LE. B. A
С	MUST BE LESS THAN B AND A
MUST BE	
C	NON-NEGATIVE. A AND B ARE IN
RADIANS.	A = 0 CORRESPONDS TO THE
C NORTH POLE AND	A = 0 CORRESPONDS TO THE
C C	B = PI CORRESPONDS TO THE
SOUTH POLE.	D II COITAGIONDO IO IIII
C	
C	* * * IMPORTANT * * *
С	
С	IF B IS EQUAL TO PI, THEN B
MUST BE	
C	COMPUTED USING THE STATEMENT
C	B = PIMACH(DUM)
C	THIS INSURES THAT B IN THE
USER'S PROGRAM	
C PROGRAM PERMITTING	IS EQUAL TO PI IN THIS
PROGRAM, PERMITTING C	CEVEDAL MECHO OF MILE INDIM
PARAMETERS THAT	SEVERAL TESTS OF THE INPUT
C C	OTHERWISE WOULD NOT BE
POSSIBLE.	OTHERWISE WOODD NOT DE
C	
C	* * * * * * * * * * *
С	
С	M
С	THE NUMBER OF GRID POINTS IN
THE INTERVAL	
С	(A,B). THE GRID POINTS IN
THE THETA-	
C	DIRECTION ARE GIVEN BY

```
THETA(I) = A + (I -
0.5) DTHETA
C
                           FOR I=1,2,...,M WHERE DTHETA
= (B-A)/M.
                           M MUST BE GREATER THAN 4.
C
С
С
                         MBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                           AT THETA = A AND THETA = B.
С
C
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
                                THETA = A AND THETA = B.
С
                                (SEE NOTES 1, 2 BELOW)
С
                           = 2 IF THE SOLUTION IS
SPECIFIED AT
                                THETA = A AND THE
DERIVATIVE OF THE
                                SOLUTION WITH RESPECT TO
THETA IS
                                SPECIFIED AT THETA = B
C
С
                                (SEE NOTES 1, 2 BELOW).
С
                           = 3 IF THE DERIVATIVE OF THE
C
SOLUTION
                                WITH RESPECT TO THETA IS
SPECIFIED
                                AT THETA = A (SEE NOTES
1, 2 BELOW)
                                AND THETA = B.
C
С
                           = 4 IF THE DERIVATIVE OF THE
SOLUTION
                                WITH RESPECT TO THETA IS
C
SPECIFIED AT
                                THETA = A (SEE NOTES 1,
2 BELOW) AND
                                THE SOLUTION IS
SPECIFIED AT THETA = B.
C
                           = 5 IF THE SOLUTION IS
UNSPECIFIED AT
```

C SOLUTION IS		THETA = $A = 0$ AND THE
C C C		SPECIFIED AT THETA = B. (SEE NOTE 2 BELOW)
C UNSPECIFIED AT	= 6	IF THE SOLUTION IS
C DERIVATIVE OF		THETA = A = 0 AND THE
C RESPECT TO THETA IS		THE SOLUTION WITH
C C C		SPECIFIED AT THETA = B (SEE NOTE 2 BELOW).
C SPECIFIED AT	= 7	IF THE SOLUTION IS
C SOLUTION IS		THETA = A AND THE
C = PI.		UNSPECIFIED AT THETA = B
C C	= 8	IF THE DERIVATIVE OF THE
SOLUTION C		WITH RESPECT TO THETA IS
SPECIFIED AT C		THETA = A (SEE NOTE 1
BELOW) C		AND THE SOLUTION IS
UNSPECIFIED AT		THETA = B = PI.
C C	= 9	IF THE SOLUTION IS
UNSPECIFIED AT		THETA = A = 0 AND THETA
= B = PI. C		
C C	NOTE IF A	1: = 0, DO NOT USE MBDCND =
1,2,3,4,7 C	OR 8,	BUT INSTEAD USE MBDCND
= 5, 6, OR 9.		
C C	NOTE IF B	2: = PI, DO NOT USE MBDCND

```
= 1, 2, 3, 4, 5,
С
                          OR 6, BUT INSTEAD USE MBDCND
= 7, 8, OR 9.
С
С
                           NOTE 3:
                           WHEN A = 0 AND/OR B = PI THE
С
ONLY
С
                           MEANINGFUL BOUNDARY CONDITION
IS
                           DU/DTHETA = 0. SEE D.
GREENSPAN,
                           'NUMERICAL ANALYSIS OF
ELLIPTIC
С
                           BOUNDARY VALUE PROBLEMS, '
С
                           HARPER AND ROW, 1965, CHAPTER
5.)
С
С
                        BDA
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                           SPECIFIES THE BOUNDARY VALUES
(IF ANY) OF
                           THE SOLUTION AT THETA = A.
C
С
С
                           WHEN MBDCND = 1, 2, OR 7,
C
                            BDA(J) = U(A,R(J)),
J=1,2,...,N.
С
С
                           WHEN MBDCND = 3, 4, OR 8,
                            BDA(J) =
(D/DTHETA) U (A,R(J)), J=1,2,...,N.
С
                          WHEN MBDCND HAS ANY OTHER
VALUE, BDA IS A
                          DUMMY VARIABLE.
C
С
С
                        BDB
С
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                           SPECIFIES THE BOUNDARY VALUES
С
OF THE
С
                           SOLUTION AT THETA = B.
С
                           WHEN MBDCND = 1, 4, OR 5,
```

```
BDB(J) = U(B,R(J)),
J=1,2,...,N.
С
С
                           WHEN MBDCND = 2,3, OR 6,
С
                             BDB(J) =
(D/DTHETA)U(B,R(J)), J=1,2,...,N.
С
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDB IS
                          A DUMMY VARIABLE.
С
С
                         C,D
С
                           THE RANGE OF R , I.E. C .LE.
R .LE. D.
                          C MUST BE LESS THAN D AND
NON-NEGATIVE.
С
                         Ν
                           THE NUMBER OF UNKNOWNS IN THE
INTERVAL
                           (C, D). THE UNKNOWNS IN THE
R-DIRECTION
                           ARE GIVEN BY R(J) = C + (J -
0.5) DR,
                           J=1,2,...,N, WHERE DR = (D-
C)/N.
                           N MUST BE GREATER THAN 4.
С
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                          AT R = C AND R = D.
С
С
С
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
                               R = C AND R = D.
C
С
                          = 2 IF THE SOLUTION IS
SPECIFIED AT
                                R = C AND THE DERIVATIVE
OF THE
С
                                SOLUTION WITH RESPECT TO
R IS
```

C		SPECIFIED AT R = D. (SEE
NOTE 1 BELOW)		
C	– 3	IF THE DERIVATIVE OF THE
SOLUTION	- 3	IF THE DERIVATIVE OF THE
C		WITH RESPECT TO R IS
SPECIFIED AT		WIII IEDIZOI IO IV IO
C		R = C AND $R = D$.
С		
C	= 4	IF THE DERIVATIVE OF THE
SOLUTION		
С		WITH RESPECT TO R IS
C		SPECIFIED AT $R = C$ AND
THE SOLUTION		
C		IS SPECIFIED AT $R = D$.
C	_	
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		R = C = 0 (SEE NOTE 2
C BELOW) AND THE		R = C = 0 (SEE NOIE 2
C AND THE		SOLUTION IS SPECIFIED AT
R = D.		SOLUTION 15 STECTFIED AT
C C		
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT	-	
С		R = C = 0 (SEE NOTE 2
BELOW)		
С		AND THE DERIVATIVE OF
THE SOLUTION		
C		WITH RESPECT TO R IS
SPECIFIED AT		
C		R = D.
C	NOTE	1
C	NOTE	
OR 9, THE	IF C	= 0 AND MBDCND = 3,6,8
C C	SYST	EM OF EQUATIONS TO BE
SOLVED IS	DIDI.	
C	SING	ULAR. THE UNIQUE
SOLUTION IS		~
С	DETE	RMINED BY EXTRAPOLATION
TO THE		
С	SPEC	IFICATION OF
U(THETA(1),C).		

```
BUT IN THESE CASES THE RIGHT
SIDE OF THE
                           SYSTEM WILL BE PERTURBED BY
THE CONSTANT
                           PERTRB.
С
С
                           NOTE 2:
                           NBDCND = 5 OR 6 CANNOT BE
С
USED WITH
                           MBDCND =1, 2, 4, 5, OR 7
                           (THE FORMER INDICATES THAT
THE SOLUTION IS
                           UNSPECIFIED AT R = 0; THE
LATTER INDICATES
                           SOLUTION IS SPECIFIED).
С
                           USE INSTEAD NBDCND = 1 \text{ OR } 2.
С
С
                        BDC
                           A ONE DIMENSIONAL ARRAY OF
LENGTH M THAT
                          SPECIFIES THE BOUNDARY VALUES
OF THE
                          SOLUTION AT R = C. WHEN
NBDCND = 1 OR 2,
                            BDC(I) = U(THETA(I), C),
I=1,2,...,M.
С
                           WHEN NBDCND = 3 OR 4,
                            BDC(I) =
(D/DR)U(THETA(I),C), I=1,2,...,M.
C
С
                          WHEN NBDCND HAS ANY OTHER
VALUE, BDC IS
                          A DUMMY VARIABLE.
С
С
                        BDD
                          A ONE-DIMENSIONAL ARRAY OF
LENGTH M THAT
                          SPECIFIES THE BOUNDARY VALUES
OF THE
                          SOLUTION AT R = D. WHEN
NBDCND = 1 OR 4,
                            BDD(I) = U(THETA(I), D),
I=1,2,...,M.
```

```
С
С
                           WHEN NBDCND = 2 \text{ OR } 3,
                             BDD(I) =
(D/DR)U(THETA(I),D), I=1,2,...,M.
С
                           WHEN NBDCND HAS ANY OTHER
VALUE, BDD IS
                          A DUMMY VARIABLE.
С
С
                         FIMBDA
                           THE CONSTANT LAMBDA IN THE
С
MODIFIED
                           HELMHOLTZ EQUATION. IF
LAMBDA IS GREATER
                           THAN 0, A SOLUTION MAY NOT
EXIST.
                           HOWEVER, HSTCSP WILL ATTEMPT
TO FIND A
                           SOLUTION.
С
С
                           A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES THE
                           VALUES OF THE RIGHT SIDE OF
THE MODIFIED
                           HELMHOLTZ EQUATION. FOR
I=1,2,\ldots,M AND
                           J=1,2,...,N
С
                                 F(I,J) =
F(THETA(I),R(J)).
С
                          F MUST BE DIMENSIONED AT
LEAST M X N.
С
                         IDIMF
                           THE ROW (OR FIRST) DIMENSION
OF THE ARRAY
                          F AS IT APPEARS IN THE
PROGRAM CALLING
                           HSTCSP. THIS PARAMETER IS
USED TO SPECIFY
C
                           THE VARIABLE DIMENSION OF F.
                           IDIMF MUST BE AT LEAST M.
```

C	
C	
(fishworkspace) variable	A fortran 90 derived TYPE
C (IISHWOIKSPACE) VALIABLE	that must be declared by the
user. The first	chae made be acciding by the
С	two declarative statements in
the user program	
С	calling HSTCSP must be:
C	HOE 6'ch
C	USE fish
C	The declarative statement
C	
С	TYPE (fishworkspace) ::
W	
C	
user program	must also be included in the
C C	The first statement makes the
fishpack module	
C	defined in the file "fish.f"
available to the	
C	user program calling HSTCSP.
The second statement c	declares a derived type
variable (defined in	acciaies a activea type
C	the module "fish.f") which is
used internally	
C	in BLKTRI to dynamically
allocate real and complex	work and used in solution
An error flag	work space used in solution.
C C	(IERROR = 20) is set if the
required work space	
С	allocation fails (for example
if N,M are too large)	
C set in the components	Real and complex values are
set in the components	of W on a initial (IFLG=0)
call to HSTCSP. These	
С	must be preserved on non-
initial calls (INTL=1)	
С	to HSTCSP. This eliminates

redundant calculations	
C	and saves compute time.
C ****	IMPORTANT! The user program
calling HSTCSP should	
C	include the statement:
C	inorado ene bedeemene.
	CALL ETGLETN (N)
C	CALL FISHFIN(W)
С	
C	after the final approximation
is generated by	
C	HSTCSP. The will deallocate
the real and complex	
C	work space of W. Failure to
include this statement	work space of w. rarrare to
	1.1 1.1
C	could result in serious
memory leakage.	
C	
С	
C	
C ON OUTPUT	F
C	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	CONTINUE THE COLOTION O(170)
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	(THETA(I),R(J)) FOR
I=1,2,,M, J=1,2,,N.	
C	
C	PERTRB
С	IF A COMBINATION OF PERIODIC,
DERIVATIVE,	
·	OD INICDECTETED DOINIDADY
C	OR UNSPECIFIED BOUNDARY
CONDITIONS IS	
C	SPECIFIED FOR A POISSON
EQUATION	
С	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
C	PERTRB IS A CONSTANT,
CALCULATED AND	
C CALCULATED AND	CLIDED COED EDOM E WILLCH
	SUBTRACTED FROM F, WHICH
ENSURES THAT A	
С	SOLUTION EXISTS. HSTCSP THEN
COMPUTES THIS	
С	SOLUTION, WHICH IS A LEAST
SQUARES SOLUTION	
~	

C APPROXIMATION.	TO THE ORIGINAL
C C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	A COLUMNAL LIBRAR MILE
C SOLUTION IS NOT	A SOLUTION; HENCE, THE
C	UNIQUE. THE VALUE OF PERTRB
SHOULD BE C	SMALL COMPARED TO THE RIGHT
SIDE F.	
C OBTAINED TO AN	OTHERWISE, A SOLUTION IS
C C	ESSENTIALLY DIFFERENT
PROBLEM.	
C BE MADE TO	THIS COMPARISON SHOULD ALWAYS
С	INSURE THAT A MEANINGFUL
SOLUTION HAS BEEN C	OBTAINED.
С	02 ·
C	IERROR AN ERROR FLAG THAT INDICATES
INVALID INPUT	AN ERROR FLAG HAI INDICATES
C	PARAMETERS. EXCEPT FOR
NUMBERS 0 AND 10,	A SOLUTION IS NOT ATTEMPTED.
C	0 110 77707
C	= 0 NO ERROR
C C C	= 0 NO ERROR = 1 A .LT. 0 OR B .GT. PI
C C C	= 1 A .LT. 0 OR B .GT. PI
C C C C	
C C C C C C	= 1 A .LT. 0 OR B .GT. PI
C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B
C C C C C C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B
C C C C C C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B = 3 MBDCND .LT. 1 OR MBDCND
C C C C C C C C C C C C C C C C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B = 3 MBDCND .LT. 1 OR MBDCND = 4 C .LT. 0 = 5 C .GE. D
C C C C C C C C C C C C C C C C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B = 3 MBDCND .LT. 1 OR MBDCND = 4 C .LT. 0
C C C C C C C C C C C C C C C C C C C	= 1 A .LT. 0 OR B .GT. PI = 2 A .GE. B = 3 MBDCND .LT. 1 OR MBDCND = 4 C .LT. 0 = 5 C .GE. D

```
C
С
                            = 8 NBDCND = 5 OR 6 AND
                                 MBDCND = 1, 2, 4, 5, OR
С
7
С
                            = 9 C .GT. 0 AND NBDCND
С
.GE. 5
C
                           = 10 ELMBDA .GT. 0
\mathsf{C}
С
С
                            = 11 IDIMF .LT. M
С
С
                           = 12 M .LT. 5
С
С
                           = 13 A = 0 AND MBDCND
=1,2,3,4,7 OR 8
С
                           = 14 B = PI AND MBDCND .LE.
6
С
С
                           = 15 A .GT. 0 AND MBDCND =
5, 6, OR 9
C
                           = 16 B .LT. PI AND MBDCND
.GE. 7
C
                           = 17 LAMBDA .NE. 0 AND
NBDCND .GE. 5
С
                           SINCE THIS IS THE ONLY MEANS
OF INDICATING
                           A POSSIBLY INCORRECT CALL TO
HSTCSP,
                            THE USER SHOULD TEST IERROR
AFTER THE CALL.
С
                        = 20 If the dynamic allocation
of real and
                              complex work space in the
derived type
                               (fishworkspace) variable W
fails (e.g.,
                              if N,M are too large for
the platform used)
```

```
С
                         W
                              The derived type
(fishworkspace) variable W
                              contains real and complex
values that must not
                              be destroyed if HSTCSP is
called again with
                              IFLG=1.
С
C I/O
                        NONE
C PRECISION
                        SINGLE
C REQUIRED LIBRARY fish.f, blktri.f, comf.f
C FILES
C
                       FORTRAN 90
C LANGUAGE
C HISTORY
                         WRITTEN BY ROLAND SWEET AT NCAR
IN 1977.
                         RELEASED ON NCAR'S PUBLIC
SOFTWARE LIBRARIES
                         IN JANUARY 1980. Revised by
John Adams in June
                         2004 using Fortan 90
dynamically allocated work
                         space and derived data types to
eliminate mixed
                         mode conflicts in the earlier
versions.
C PORTABILITY
                        FORTRAN 90
C ALGORITHM
                         THIS SUBROUTINE DEFINES THE
FINITE-DIFFERENCE
                         EQUATIONS, INCORPORATES
BOUNDARY DATA, ADJUSTS
                         THE RIGHT SIDE WHEN THE SYSTEM
IS SINGULAR
                         AND CALLS BLKTRI WHICH SOLVES
THE LINEAR
                         SYSTEM OF EQUATIONS.
```

C	
C TIMING	EOD IADCE MAND N. HIE
OPERATION COUNT IS	FOR LARGE M AND N, THE
C C	ROUGHLY PROPORTIONAL TO
M*N*LOG2(N). THE	
C	TIMING ALSO DEPENDS ON INPUT
PARAMETER INTL.	
C	
C ACCURACY	THE SOLUTION PROCESS EMPLOYED
RESULTS IN	
C	A LOSS OF NO MORE THAN FOUR
SIGNIFICANT	
C	DIGITS FOR N AND M AS LARGE AS
64.	MODEL DEED THEODIGHTON ADOLE
C ACCURACY	MORE DETAILED INFORMATION ABOUT
C	CAN BE FOUND IN THE
DOCUMENTATION FOR	CAN DE LOOND IN THE
C	SUBROUTINE BLKTRI WHICH IS THE
ROUTINE	
C	SOLVES THE FINITE DIFFERENCE
EQUATIONS.	
C	
C REFERENCES	P.N. SWARZTRAUBER, "A DIRECT
METHOD FOR	
C	THE DISCRETE SOLUTION OF
SEPARABLE ELLIPTIC	EOID ETONO!!
C	EQUATIONS", SIAM J. NUMER. ANAL. 11(1974),
PP. 1136-1150.	SIAT O. NOPER. ANAL. II(15/4),
C	
C	U. SCHUMANN AND R. SWEET, "A
DIRECT METHOD FOR	,
C	THE SOLUTION OF POISSON'S
EQUATION WITH NEUMANN	
C	BOUNDARY CONDITIONS ON A
STAGGERED GRID OF	
C (1.07.6)	ARBITRARY SIZE," J. COMP. PHYS.
20 (1976),	DD 171 100
C	PP. 171-182.

HSTCYL

```
file hstcyl.txt (documentation for the FISHPACK
solver HSTCYL)
С
                         copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
                             all rights reserved
C
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
```

```
C
               John Adams, Paul Swarztrauber and Roland
Sweet
С
C
                                     of
С
                the National Center for Atmospheric
Research
                        Boulder, Colorado (80307)
U.S.A.
C
С
                           which is sponsored by
С
С
                      the National Science Foundation
С
С
      SUBROUTINE HSTCYL
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                         ELMBDA, F, IDIMF, PERTRB, IERROR)
С
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N)
C ARGUMENTS
C LATEST REVISION
                         June 2004
C PURPOSE
                         SOLVES THE STANDARD FIVE-POINT
FINITE
С
                          DIFFERENCE APPROXIMATION ON A
STAGGERED
                          GRID TO THE MODIFIED HELMHOLTZ
```

```
EQUATION
                          IN CYLINDRICAL COORDINATES.
С
THIS EQUATION
C
С
                            (1/R) (D/DR) (R(DU/DR)) +
(D/DZ)(DU/DZ)
C
С
                            + LAMBDA* (1/R**2)*U = F(R,Z)
С
C
                          IS A TWO-DIMENSIONAL MODIFIED
HELMHOLTZ
                         EQUATION RESULTING FROM THE
FOURIER TRANSFORM
                         OF A THREE-DIMENSIONAL POISSON
EQUATION.
С
C USAGE
                          CALL HSTCYL
(A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
С
                                        PERTRB, IERROR)
C
C ARGUMENTS
C ON INPUT
                          A,B
С
                            THE RANGE OF R, I.E. A .LE. R
C
.LE. B.
                            A MUST BE LESS THAN B AND A
MUST BE
                           BE NON-NEGATIVE.
С
С
С
                          M
                            THE NUMBER OF GRID POINTS IN
С
THE INTERVAL
                            (A,B). THE GRID POINTS IN
THE R-DIRECTION
С
                            R-DIRECTION ARE GIVEN BY
                            R(I) = A + (I-0.5)DR FOR
C
I=1,2,...,M
                            WHERE DR = (B-A)/M.
C
                            M MUST BE GREATER THAN 2.
С
С
С
                          MBDCND
                            INDICATES THE TYPE OF
```

BOUNDARY CONDITIONS		
C	AT R	= A AND R = B.
C	= 1	IF THE SOLUTION IS
SPECIFIED AT R = A		/
C B.		(SEE NOTE BELOW) AND R =
C		
C	= 2	IF THE SOLUTION IS
SPECIFIED AT R = A		(SEE NOTE BELOW) AND THE
DERIVATIVE		(ODD NOTE DEBOW) TEND THE
C DESCRIPTION D. T.S.		OF THE SOLUTION WITH
RESPECT TO R IS		SPECIFIED AT R = B.
C		
C SOLUTION	= 3	IF THE DERIVATIVE OF THE
C		WITH RESPECT TO R IS
SPECIFIED AT		
C AND $R = B$.		R = A (SEE NOTE BELOW)
C C		
С	= 4	IF THE DERIVATIVE OF THE
SOLUTION C		WITH RESPECT TO R IS
SPECIFIED AT		WIII IMELIECI IO IC ID
C		R = A (SEE NOTE BELOW)
AND THE C		SOLUTION IS SPECIFIED AT
R = B.		
C		TE MUE COLUMNON TO
C UNSPECIFIED AT	= 5	IF THE SOLUTION IS
С		R = A = 0 AND THE
SOLUTION IS C		SPECIFIED AT R = B.
C		SPECIFIED AT K - D.
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT		R = A = 0 AND THE
DERIVATIVE OF THE		7. 11 0 1140 1110
C		SOLUTION WITH RESPECT TO
R IS SPECIFIED		

```
AT R = B.
С
С
                            NOTE:
C
                            IF A = 0, DO NOT USE MBDCND =
1,2,3, OR 4,
                            BUT INSTEAD USE MBDCND = 5 OR
6.
                            THE RESULTING APPROXIMATION
GIVES THE ONLY
                            MEANINGFUL BOUNDARY
CONDITION,
С
                            I.E. DU/DR = 0.
                            (SEE D. GREENSPAN,
'INTRODUCTORY NUMERICAL
                            ANALYSIS OF ELLIPTIC BOUNDARY
VALUE
                           PROBLEMS, ' HARPER AND ROW,
1965, CHAPTER 5.)
С
                         BDA
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                            SPECIFIES THE BOUNDARY VALUES
(IF ANY)
                            OF THE SOLUTION AT R = A.
С
С
                            WHEN MBDCND = 1 \text{ OR } 2,
С
                              BDA(J) = U(A,Z(J)),
J=1,2,...,N.
С
С
                            WHEN MBDCND = 3 OR 4,
                              BDA(J) = (D/DR)U(A,Z(J)),
С
J=1,2,...,N.
С
                            WHEN MBDCND = 5 OR 6, BDA IS
A DUMMY
С
                            VARIABLE.
С
С
                         BDB
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
                            SOLUTION AT R = B.
```

```
С
                           WHEN MBDCND = 1,4,OR 5,
C
                             BDB(J) = U(B, Z(J)),
J=1,2,...,N.
С
                           WHEN MBDCND = 2,3, OR 6,
                             BDB(J) = (D/DR)U(B,Z(J)),
C
J=1,2,...,N.
С
                         C,D
С
                           THE RANGE OF Z, I.E. C .LE. Z
.LE. D.
                           C MUST BE LESS THAN D.
C
С
С
                         Ν
C
                           THE NUMBER OF UNKNOWNS IN THE
INTERVAL
                           (C,D). THE UNKNOWNS IN THE
Z-DIRECTION
                           ARE GIVEN BY Z(J) = C + (J -
0.5) DZ,
                           J=1,2,...,N, WHERE DZ = (D-
C
C)/N.
С
                           N MUST BE GREATER THAN 2.
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                           AT Z = C AND Z = D.
С
С
                           = 0 IF THE SOLUTION IS
PERIODIC IN Z, I.E.
                                U(I,J) = U(I,N+J).
C
С
                           = 1 IF THE SOLUTION IS
SPECIFIED AT Z = C
С
                                AND Z = D.
С
                           = 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
С
SOLUTION WITH
                                 RESPECT TO Z IS
SPECIFIED AT Z = C
                                 AND Z = D.
С
С
                           = 4 IF THE DERIVATIVE OF THE
SOLUTION WITH
                                 RESPECT TO Z IS
SPECIFIED AT Z = C AND
                                 THE SOLUTION IS
SPECIFIED AT Z = D.
С
                          BDC
C
                            A ONE DIMENSIONAL ARRAY OF
LENGTH M THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
                            SOLUTION AT Z = C.
С
С
                            WHEN NBDCND = 1 \text{ OR } 2,
C
                              BDC(I) = U(R(I),C),
I=1,2,...,M.
С
                            WHEN NBDCND = 3 OR 4,
                              BDC(I) = (D/DZ)U(R(I),C),
I=1,2,...,M.
                            WHEN NBDCND = 0, BDC IS A
DUMMY VARIABLE.
C
С
                          BDD
                            A ONE-DIMENSIONAL ARRAY OF
LENGTH M THAT
C
                            SPECIFIES THE BOUNDARY VALUES
OF THE
                            SOLUTION AT Z = D.
C
С
С
                            WHEN NBDCND = 1 \text{ OR } 4,
                              BDD(I) = U(R(I),D),
I=1,2,...,M.
C
                            WHEN NBDCND = 2 \text{ OR } 3,
```

```
BDD(I) = (D/DZ)U(R(I),D),
I=1,2,...,M.
С
C
                           WHEN NBDCND = 0, BDD IS A
DUMMY VARIABLE.
С
                        FILMBDA
С
                           THE CONSTANT LAMBDA IN THE
MODIFIED
                           HELMHOLTZ EQUATION. IF
LAMBDA IS GREATER
                           THAN O, A SOLUTION MAY NOT
EXIST.
                           HOWEVER, HSTCYL WILL ATTEMPT
TO FIND A
                           SOLUTION. LAMBDA MUST BE
ZERO WHEN
                           MBDCND = 5 OR 6.
C
С
С
C
                           A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                           THE VALUES OF THE RIGHT SIDE
C
OF THE
С
                           MODIFIED HELMHOLTZ EQUATION.
                           FOR I=1,2,...,M AND
C
J=1,2,...,N
                            F(I,J) = F(R(I),Z(J)).
C
                           F MUST BE DIMENSIONED AT
С
LEAST M X N.
C
С
                        IDIMF
С
                           THE ROW (OR FIRST) DIMENSION
OF THE ARRAY
                          F AS IT APPEARS IN THE
PROGRAM CALLING
                          HSTCYL. THIS PARAMETER IS
USED TO SPECIFY
                          THE VARIABLE DIMENSION OF F.
IDIMF MUST
                          BE AT LEAST M.
С
C ON OUTPUT
```

C	F CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	CONTINUE THE COLOTION O(170)
	DIEGEDENICE ADDOVINATION FOR
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	$(R(I),Z(J))$ FOR $I=1,2,\ldots,M$,
J=1,2,,N.	
C	
С	PERTRB
C	IF A COMBINATION OF PERIODIC,
DERIVATIVE,	
·	OD INICDECTETED DOINIDADY
C	OR UNSPECIFIED BOUNDARY
CONDITIONS IS	
C	SPECIFIED FOR A POISSON
EQUATION	
С	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
С	PERTRB IS A CONSTANT,
CALCULATED AND	,
C	SUBTRACTED FROM F, WHICH
	SOBINACIED FROM F, WILLON
ENSURES THAT A	
C	SOLUTION EXISTS. HSTCYL THEN
COMPUTES	
C	THIS SOLUTION, WHICH IS A
LEAST SQUARES	
C	SOLUTION TO THE ORIGINAL
APPROXIMATION.	
C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	11110 001011011 1100 1111
	A COLUMNOM LIENOR MILE
C	A SOLUTION; HENCE, THE
SOLUTION IS NOT	
C	UNIQUE. THE VALUE OF PERTRB
SHOULD BE	
С	SMALL COMPARED TO THE RIGHT
SIDE F.	
С	OTHERWISE, A SOLUTION IS
OBTAINED TO AN	officially if colorion is
C	ESSENTIALLY DIFFERENT
	ESSENITALLI DIFFERENI
PROBLEM.	THE COMPANION CHANGE TO THE COMPANION OF
C	THIS COMPARISON SHOULD ALWAYS
BE MADE TO	
С	INSURE THAT A MEANINGFUL
SOLUTION HAS BEEN	
C	OBTAINED.

C	
	THDDOD
С	IERROR
С	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS. EXCEPT TO NUMBERS
0 AND 11,	
C	A SOLUTION IS NOT ATTEMPTED.
C	
C	= 0 NO ERROR
	- U NO ERROR
С	
C	= 1 A .LT. 0
С	
C	= 2 A .GE. B
C	
C	= 3 MBDCND .LT. 1 OR MBDCND
.GT. 6	3 IEDOND .III. I ON IEDOND
C	4 0 07 7
С	= 4 C .GE. D
C	
C	= 5 N .LE. 2
C	
C	= 6 NBDCND .LT. 0 OR NBDCND
.GT. 4	0 1.2501.2 1.21 0 01. 1.2201.2
C C	
C	= 7 A $=$ 0 AND MBDCND $=$
1,2,3, OR 4	
C	
C	= 8 A .GT. 0 AND MBDCND
.GE. 5	
C	
C	= 9 M .LE. 2
	- J M . DE . Z
C	10 TDTM
С	= 10 IDIMF .LT. M
C	
С	= 11 LAMBDA .GT. 0
C	
C	= 12 A=0, MBDCND .GE. 5,
ELMBDA .NE. 0	
C	
	OTNOR BUTO TO BUE ONTY MEANS
С	SINCE THIS IS THE ONLY MEANS
OF INDICATING	
С	A POSSIBLY INCORRECT CALL TO
HSTCYL, THE	
C	USER SHOULD TEST IERROR AFTER

```
THE CALL.
С
                           = 20 If the dynamic
allocation of real and
                                complex work space
required for solution
                                fails (for example if
N,M are too large
С
                                for your computer)
C I/O
                        NONE
C PRECISION
                        SINGLE
C REQUIRED LIBRARY
fish.f,comf.f,genbun.f,gnbnaux.f,poistg.f
C FILES
C LANGUAGE
                        FORTRAN 90
C HISTORY
                        WRITTEN BY ROLAND SWEET AT NCAR
IN 1977.
                        RELEASED ON NCAR'S PUBLIC
SOFTWARE LIBRARIES
                         IN JANUARY 1980.
                         Revised in June 2004 by John
Adams using
                        Fortran 90 dynamically
allocated work space.
C PORTABILITY
                        FORTRAN 90
C ALGORITHM
                         THIS SUBROUTINE DEFINES THE
FINITE-DIFFERENCE
                         EQUATIONS, INCORPORATES
BOUNDARY DATA, ADJUSTS
                         THE RIGHT SIDE WHEN THE SYSTEM
С
IS SINGULAR AND
                         CALLS EITHER POISTG OR GENBUN
WHICH SOLVES THE
                         LINEAR SYSTEM OF EQUATIONS.
C
C TIMING
                         FOR LARGE M AND N, THE
OPERATION COUNT
                         IS ROUGHLY PROPORTIONAL TO
```

```
M*N*LOG2(N).
С
C ACCURACY
                     THE SOLUTION PROCESS RESULTS IN
A LOSS
С
                      OF NO MORE THAN FOUR
SIGNIFICANT DIGITS
                      FOR N AND M AS LARGE AS 64.
С
                      MORE DETAILED INFORMATION ABOUT
ACCURACY
                      CAN BE FOUND IN THE
DOCUMENTATION FOR
                      SUBROUTINE POISTG WHICH IS THE
ROUTINE THAT
                      ACTUALLY SOLVES THE FINITE
DIFFERENCE
                      EQUATIONS.
C REFERENCES
                     U. SCHUMANN AND R. SWEET, "A
DIRECT METHOD FOR
                      THE SOLUTION OF POISSON'S
EQUATION WITH NEUMANN
                      BOUNDARY CONDITIONS ON A
STAGGERED GRID OF
                      ARBITRARY SIZE," J. COMP. PHYS.
20 (1976),
                      PP. 171-182.
```

HSTPLR

```
copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
С
Research
                             all rights reserved
С
                            FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
```

```
С
                      Boulder, Colorado (80307)
U.S.A.
C
С
                           which is sponsored by
С
С
                     the National Science Foundation
С
С
     SUBROUTINE HSTPLR
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
C +
                         ELMBDA, F, IDIMF, PERTRB, IERROR)
С
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N)
C ARGUMENTS
C LATEST REVISION
                         June 2004
C
C PURPOSE
                         SOLVES THE STANDARD FIVE-POINT
FINITE
                         DIFFERENCE APPROXIMATION ON A
STAGGERED
                         GRID TO THE HELMHOLTZ EQUATION
IN POLAR
С
                          COORDINATES. THE EQUATION IS
С
С
                             (1/R) (D/DR) (R(DU/DR)) +
(1/R**2) (D/DTHETA) (DU/DTHETA) +
                             LAMBDA*U = F(R, THETA)
C
С
                          CALL HSTPLR
C USAGE
(A,B,M,MBDCND,BDA,BDB,C,D,N,
NBDCND, BDC, BDD, ELMBDA, F,
```

IDIMF, PERTRB, IERROR)	
C	
C ARGUMENTS	
C ON INPUT	A , B
C	
C	THE RANGE OF R, I.E. A .LE. R
.LE. B.	
С	A MUST BE LESS THAN B AND A
MUST BE	
С	NON-NEGATIVE.
C	
C	M
C	THE NUMBER OF GRID POINTS IN
THE INTERVAL	THE NOMBER OF GRID FOIRIS IN
	(A D) MILE COID DOINING IN
C	(A,B). THE GRID POINTS IN
THE R-DIRECTION	
C	ARE GIVEN BY $R(I) = A + (I -$
0.5) DR FOR	- 1 0
C	$I=1,2,\ldots,M$ WHERE DR = (B-
A) /M.	
C	M MUST BE GREATER THAN 2.
C	
C	MBDCND
C	INDICATES THE TYPE OF
BOUNDARY CONDITIONS	
C	AT $R = A$ AND $R = B$.
C	
С	= 1 IF THE SOLUTION IS
SPECIFIED AT R = A	
С	AND $R = B$.
С	
C	= 2 IF THE SOLUTION IS
SPECIFIED AT R = A	
C	AND THE DERIVATIVE OF
THE SOLUTION	AND THE DERIVATIVE OF
C	WITH RESPECT TO R IS
	WIIN RESPECT TO K IS
SPECIFIED AT R = B.	(CDD NOWD 1 DDION)
C	(SEE NOTE 1 BELOW)
C	^
C	= 3 IF THE DERIVATIVE OF THE
SOLUTION	
C	WITH RESPECT TO R IS
SPECIFIED AT	
C	R = A (SEE NOTE 2 BELOW)

AND $R = B$.		
C		
C	= 4	IF THE DERIVATIVE OF THE
SOLUTION		MITMLE DECDECE MO D. T.C.
C CDECLETED AD		WITH RESPECT TO R IS
SPECIFIED AT		SPECIFIED AT R = A (SEE
NOTE 2 BELOW)		SPECIFIED AT R - A (SEE
C BELOW)		AND THE SOLUTION IS
SPECIFIED AT R = B.		
C		
C		
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		
С		R = A = 0 AND THE
SOLUTION IS		
C		SPECIFIED AT $R = B$.
C		
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT		_
C		R = A = 0 AND THE
DERIVATIVE OF THE		
C D TG GDEGTETED		SOLUTION WITH RESPECT TO
R IS SPECIFIED		AT $R = B$.
C		AI K - B.
C	NOTE	1.
C		= 0, MBDCND $= 2$, AND
NBDCND = 0 OR 3,		-, -=
C	THE S	SYSTEM OF EQUATIONS TO BE
SOLVED IS		
C	SING	ULAR. THE UNIQUE
SOLUTION IS		
С	IS D	ETERMINED BY
EXTRAPOLATION TO THE		
C	SPEC	IFICATION OF
U(0,THETA(1)).		
C	BUT 1	IN THIS CASE THE RIGHT
SIDE OF THE	CVCIII	EM VITT DE DEDMIDDED DV
C THE CONSTANT	21211	EM WILL BE PERTURBED BY
C C	PERTI	RR
C	11/11	····· •
C	NOTE	2:
-		

```
IF A = 0, DO NOT USE MBDCND =
3 OR 4,
С
                            BUT INSTEAD USE MBDCND =
1,2,5, OR 6.
С
                          BDA
C
                            A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                            SPECIFIES THE BOUNDARY VALUES
(IF ANY) OF
                            THE SOLUTION AT R = A.
C
С
                            WHEN MBDCND = 1 \text{ OR } 2,
С
                              BDA(J) = U(A, THETA(J)),
J=1,2,...,N.
C
С
                            WHEN MBDCND = 3 OR 4,
С
                              BDA(J) =
(D/DR)U(A,THETA(J)),
                              J=1,2,...,N.
C
С
C
                            WHEN MBDCND = 5 OR 6, BDA IS
A DUMMY
С
                            VARIABLE.
С
С
                          BDB
                            A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
C
                            SOLUTION AT R = B.
С
С
                            WHEN MBDCND = 1,4, OR 5,
                              BDB(J) = U(B, THETA(J)),
J=1,2,...,N.
С
С
                            WHEN MBDCND = 2,3, OR 6,
C
                              BDB(J) =
(D/DR)U(B,THETA(J)),
                              J=1,2,...,N.
C
С
С
                          C,D
С
                            THE RANGE OF THETA, I.E. C
.LE. THETA .LE. D.
```

```
C MUST BE LESS THAN D.
С
C
                           THE NUMBER OF UNKNOWNS IN THE
INTERVAL
                           (C,D). THE UNKNOWNS IN THE
THETA-
                          DIRECTION ARE GIVEN BY
THETA(J) = C +
                          (J-0.5)DT, J=1,2,...,N,
WHERE
                          DT = (D-C)/N. N MUST BE
GREATER THAN 2.
С
                        NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                          AT THETA = C AND THETA = D.
С
                          = 0 IF THE SOLUTION IS
PERIODIC IN THETA,
                                I.E. U(I,J) = U(I,N+J).
C
                          = 1 IF THE SOLUTION IS
SPECIFIED AT
                                THETA = C AND THETA = D
С
                                (SEE NOTE BELOW).
С
                          = 2 IF THE SOLUTION IS
SPECIFIED AT
                                THETA = C AND THE
DERIVATIVE OF THE
                                SOLUTION WITH RESPECT TO
THETA IS
                                SPECIFIED AT THETA = D
С
                                (SEE NOTE BELOW).
С
С
                          = 3 IF THE DERIVATIVE OF THE
SOLUTION
                               WITH RESPECT TO THETA IS
SPECIFIED
С
                               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
С
                                 (SEE NOTE BELOW).
С
С
                           NOTE:
                           WHEN NBDCND = 1, 2, OR 4, DO
С
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.
С
                         BDC
C
                           A ONE DIMENSIONAL ARRAY OF
LENGTH M THAT
                           SPECIFIES THE BOUNDARY VALUES
OF THE
                           SOLUTION AT THETA = C.
С
С
                           WHEN NBDCND = 1 \text{ OR } 2,
С
                             BDC(I) = U(R(I),C),
I=1,2,...,M.
C
С
                           WHEN NBDCND = 3 OR 4,
                             BDC(I) =
(D/DTHETA)U(R(I),C),
С
                             I=1,2,...,M.
С
С
                           WHEN NBDCND = 0, BDC IS A
DUMMY VARIABLE.
C
C
                         BDD
С
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH M THAT
                           SPECIFIES THE BOUNDARY VALUES
```

```
OF THE
                            SOLUTION AT THETA = D.
С
C
С
                            WHEN NBDCND = 1 \text{ OR } 4,
С
                              BDD(I) = U(R(I),D),
I=1,2,...,M.
C
С
                            WHEN NBDCND = 2 \text{ OR } 3,
С
                             BDD(I)
= (D/DTHETA) U (R(I), D), I=1, 2, ..., M.
С
                            WHEN NBDCND = 0, BDD IS A
DUMMY VARIABLE.
С
                          ELMBDA
C
                            THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                           EQUATION. IF LAMBDA IS
GREATER THAN 0,
                           A SOLUTION MAY NOT EXIST.
HOWEVER, HSTPLR
                           WILL ATTEMPT TO FIND A
SOLUTION.
С
С
                         F
C
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES THE
                            VALUES OF THE RIGHT SIDE OF
THE HELMHOLTZ
С
                            EQUATION.
С
С
                            FOR I=1,2,\ldots,M AND
J=1,2,...,N
                             F(I,J) = F(R(I),THETA(J)).
С
С
                           F MUST BE DIMENSIONED AT
LEAST M X N.
C
С
                          IDIMF
                           THE ROW (OR FIRST) DIMENSION
C
OF THE ARRAY
                           F AS IT APPEARS IN THE
PROGRAM CALLING
                            HSTPLR. THIS PARAMETER IS
```

USED TO SPECIFY	
C	THE VARIABLE DIMENSION OF F.
C	IDIMF MUST BE AT LEAST M.
C	
С	
C ON OUTPUT	
С	
С	F
С	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	(R(I), THETA(J)) FOR
$I=1,2,\ldots,M$	
C	J=1,2,,N.
С	
С	PERTRB
С	IF A COMBINATION OF PERIODIC,
DERIVATIVE,	·
C	OR UNSPECIFIED BOUNDARY
CONDITIONS IS	
С	SPECIFIED FOR A POISSON
EQUATION	
C	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
С	PERTRB IS A CONSTANT
CALCULATED AND	
С	SUBTRACTED FROM F, WHICH
ENSURES THAT A	·
С	SOLUTION EXISTS. HSTPLR THEN
COMPUTES THIS	
С	SOLUTION, WHICH IS A LEAST
SQUARES SOLUTION	·
C	TO THE ORIGINAL
APPROXIMATION.	
C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	
C	A SOLUTION; HENCE, THE
SOLUTION IS NOT	
C	UNIQUE. THE VALUE OF PERTRB
SHOULD BE	~
C	SMALL COMPARED TO THE RIGHT
SIDE F.	
C	OTHERWISE, A SOLUTION IS

OBTAINED TO AN	
C	ESSENTIALLY DIFFERENT
PROBLEM.	MILLO COMPADICONI GLIGILID ALLIAVO
C DE MADE EO	THIS COMPARISON SHOULD ALWAYS
BE MADE TO	THOUTE BUILD A MEANITHCEUR
C COLUMNON HAS DEEN	INSURE THAT A MEANINGFUL
SOLUTION HAS BEEN	
C	OBTAINED.
C	THDDOD
C	IERROR
C TABLED TARRES	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C O TAND 11	PARAMETERS. EXCEPT TO NUMBERS
0 AND 11,	A COLUMNOM TO NOW ASSESSED
C	A SOLUTION IS NOT ATTEMPTED.
C	- 0 NO EDDOD
C	= 0 NO ERROR
C	= 1 A .LT. 0
C	= I A .LI. U
	= 2 A .GE. B
C	- Z A .GL. B
C	= 3 MBDCND .LT. 1 OR MBDCND
.GT. 6	- 3 MBDCND .LI. I OK MBDCND
.G1. 0	
C	= 4 C .GE. D
C	- 4 C .GE. D
C	= 5 N .LE. 2
C	— J IN . LIE. Z
C	= 6 NBDCND .LT. 0 OR NBDCND
.GT. 4	- 0 NBDCND .HI. 0 OK NBDCND
.G1. 4	
C	= 7 A $=$ 0 AND MBDCND $=$ 3 OR
4	- / A - 0 AND MBDCND - 3 OK
C	
C	
	= 8 A .GT. 0 AND MBDCND
.GE. 5	
C	_ O MDDOND OF E AND
C NEDGNE NE O OF 3	= 9 MBDCND .GE. 5 AND
NBDCND .NE. 0 OR 3	
C	- 10 TDIME IT M
C	= 10 IDIMF .LT. M
C	- 11 IAMDDA CELO
С	= 11 LAMBDA .GT. 0

```
С
                          = 12 M .LE. 2
С
                          = 20 If the dynamic
allocation of real and
                               complex work space
required for solution
                               fails (for example if
N,M are too large
                               for your computer)
С
C
                         SINCE THIS IS THE ONLY MEANS
OF INDICATING
                         A POSSIBLY INCORRECT CALL TO
HSTPLR, THE
                        USER SHOULD TEST IERROR AFTER
THE CALL.
С
С
C I/O
                       NONE
C
C PRECISION
                       SINGLE
C
C REQUIRED FILES
fish.f,comf.f,genbun.f,gnbnaux.f,poistg.f
C
C LANGUAGE
                       FORTRAN 90
C HISTORY
                    WRITTEN BY ROLAND SWEET AT NCAR
IN 1977.
                       RELEASED ON NCAR'S PUBLIC
SOFTWARE LIBRARIES
                        IN JANUARY 1980.
С
                       Revised in June 2004 by John
Adams using
                       Fortran 90 dynamically
allocated work space.
C PORTABILITY FORTRAN 90
C ALGORITHM
                       THIS SUBROUTINE DEFINES THE
FINITE-
                        DIFFERENCE EQUATIONS,
INCORPORATES BOUNDARY
```

C WHEN THE SYSTEM	DATA, ADJUSTS THE RIGHT SIDE
C	IS SINGULAR AND CALLS EITHER
POISTG OR GENBUN	
C	WHICH SOLVES THE LINEAR SYSTEM
OF EQUATIONS.	
C	
C TIMING	FOR LARGE M AND N, THE
OPERATION COUNT	,
С	IS ROUGHLY PROPORTIONAL TO
M*N*LOG2(N).	
C	
C ACCURACY	THE SOLUTION PROCESS EMPLOYED
RESULTS IN	
C	A LOSS OF NO MORE THAN FOUR
SIGNIFICANT	
C	DIGITS FOR N AND M AS LARGE AS
64.	
С	MORE DETAILED INFORMATION ABOUT
ACCURACY	
C	CAN BE FOUND IN THE
DOCUMENTATION FOR	
C	ROUTINE POISTG WHICH IS THE
ROUTINE THAT	
C	ACTUALLY SOLVES THE FINITE
DIFFERENCE	
C	EQUATIONS.
C	II COULTMANNI AND D OFTEN UA
C REFERENCES	U. SCHUMANN AND R. SWEET, "A
DIRECT METHOD	EOD MIE COLUMION OF DOISCONIS
C EQUATION WITH	FOR THE SOLUTION OF POISSON'S
EQUATION WITH	NEUMANN BOUNDARY CONDITIONS ON
A STAGGERED	NEOMANN BOUNDARI CONDITIONS ON
C	GRID OF ARBITRARY SIZE," J.
COMP. PHYS.	ONID OF ARBITRANT STAB, 0.
C C	20(1976), PP. 171-182.

HSTSSP

```
С
      file hstssp.txt (documentation for the FISHPACK
solver HSTSSP)
                         copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
                             all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
*
               John Adams, Paul Swarztrauber and Roland
С
```

```
Sweet
С
С
                                     of
С
С
                the National Center for Atmospheric
Research
                      Boulder, Colorado (80307)
U.S.A.
С
С
                           which is sponsored by
С
*
С
                     the National Science Foundation
C
С
      SUBROUTINE HSTSSP
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                         ELMBDA, F, IDIMF, PERTRB, IERROR)
С
C
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N)
C ARGUMENTS
C LATEST REVISION June 2004
C PURPOSE
                         SOLVES THE STANDARD FIVE-POINT
FINITE
                         DIFFERENCE APPROXIMATION ON A
STAGGERED GRID
                          TO THE HELMHOLTZ EQUATION IN
SPHERICAL
                          COORDINATES AND ON THE SURFACE
```

```
OF THE UNIT
                          SPHERE (RADIUS OF 1). THE
С
EQUATION IS
С
С
(1/SIN(THETA)) (D/DTHETA) (SIN(THETA)
                            (DU/DTHETA)) +
(1/SIN(THETA)**2)
С
                            (D/DPHI) (DU/DPHI) + LAMBDA*U
= F(THETA, PHI)
C
C
                         WHERE THETA IS COLATITUDE AND
PHI IS
С
                          LONGITUDE.
С
C USAGE
                          CALL HSTSSP
(A,B,M,MBDCND,BDA,BDB,C,D,N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
С
                                       PERTRB, IERROR)
С
C
C ARGUMENTS
C ON INPUT
С
С
                          A,B
                            THE RANGE OF THETA
(COLATITUDE),
                            I.E. A .LE. THETA .LE. B.
С
                            A MUST BE LESS THAN B AND A
MUST BE
                           NON-NEGATIVE. A AND B ARE IN
C
RADIANS.
                            A = 0 CORRESPONDS TO THE
NORTH POLE AND
С
                            B = PI CORRESPONDS TO THE
SOUTH POLE.
C
С
С
                              * * * IMPORTANT * * *
С
С
                            IF B IS EQUAL TO PI, THEN B
MUST BE
                            COMPUTED USING THE STATEMENT
```

C	B = PIMACH (DUM)
C	THIS INSURES THAT B IN THE
USER"S PROGRAM	
C	IS EQUAL TO PI IN THIS
PROGRAM WHICH	PERMITS SEVERAL TESTS OF THE
CINPUT	LEVALLE SEARWAY 15212 OF THE
C	PARAMETERS THAT OTHERWISE
WOULD NOT BE	
C	POSSIBLE.
C	* * * * * * * * * * *
C	M
C	THE NUMBER OF GRID POINTS IN
THE INTERVAL	/
C THE THETA	(A,B). THE GRID POINTS IN
C C	DIRECTION ARE GIVEN BY
C	THETA(I) = A + $(I-0.5)$ DTHETA
C	FOR $I=1,2,,M$ WHERE DTHETA
= (B-A)/M.	M MIGH DE CDEAMED MIANI O
C	M MUST BE GREATER THAN 2.
C	MBDCND
С	INDICATES THE TYPE OF
BOUNDARY CONDITIONS	3.5
C	AT THETA = A AND THETA = B.
C	= 1 IF THE SOLUTION IS
SPECIFIED AT	
С	THETA = A AND THETA = B.
C	(SEE NOTE 3 BELOW)
C	= 2 IF THE SOLUTION IS
SPECIFIED AT	
С	THETA = A AND THE
DERIVATIVE OF THE	COLUMNON THE DESPECT TO
C THETA IS	SOLUTION WITH RESPECT TO
C	SPECIFIED AT THETA = B
C	(SEE NOTES 2 AND 3
	(SEE NOTES 2 AND 3

С	= 3	IF THE DERIVATIVE OF THE
SOLUTION	ΤΛΤ	ITH RESPECT TO THETA IS
SPECIFIED	VV	IIII AESPECI TO THETA IS
С	А	T THETA = A
С	(SEE NOTES 1, 2 BELOW) AND
THETA = B.		
C	_ 1	IF THE DERIVATIVE OF THE
SOLUTION	- 4	IF THE DERIVATIVE OF THE
C		WITH RESPECT TO THETA IS
SPECIFIED		
C		AT THETA = A
С		(SEE NOTES 1 AND 2
BELOW) AND THE		
C THETA = B.		SOLUTION IS SPECIFIED AT
C		
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		
С		THETA = $A = 0$ AND THE
SOLUTION IS		
C		SPECIFIED AT THETA = B.
C		(SEE NOTE 3 BELOW)
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT	· ·	
С		THETA = $A = 0$ AND THE
DERIVATIVE		
C		OF THE SOLUTION WITH
RESPECT TO THETA		IS SPECIFIED AT THETA =
В		IS SPECIFIED AT THETA -
C		(SEE NOTE 2 BELOW).
C		· ·
С	= 7	IF THE SOLUTION IS
SPECIFIED AT		
C		THETA = A AND THE
SOLUTION IS C		UNSPECIFIED AT THETA = B
= PI.		ONOTHOTITID AT THEIR - D
C		(SEE NOTE 3 BELOW)
С		
C	= 8	IF THE DERIVATIVE OF THE

SOLUTION C SPECIFIED AT C SPECIFIED AT C BELOW) C UNSPECIFIED AT C C UNSPECIFIED AT C C C C C S S S S S S S S S S S S S S		
SPECIFIED AT C BELOW) C UNSPECIFIED AT C C UNSPECIFIED AT C C C UNSPECIFIED AT C C C C C C C C C C C C C C C C C C C	SOLUTION	
C THETA = A (SEE NOTE 1 BELOW) C AND THE SOLUTION IS UNSPECIFIED AT C THETA = B = PI. C E P IF THE SOLUTION IS UNSPECIFIED AT C THETA = A = 0 AND THETA E B = PI. C NOTE 1: C NOTE 1: C IF A = 0, DO NOT USE MBDCND = 5, 6, OR 9. C NOTE 2: C NOTE 3: C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C THETA = O AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS EXTRAPOLATION TO THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	WITH RESPECT TO THETA IS
BELOW) C C C C C C C C C C C C C C C C C C C	SPECIFIED AT	
C AND THE SOLUTION IS UNSPECIFIED AT C = 9 IF THE SOLUTION IS UNSPECIFIED AT C = B = PI. C	C	THETA = A (SEE NOTE 1
UNSPECIFIED AT C THETA = B = PI. C THETA = A = 0 AND THETA THETA = D AND THETA THETA = D AND THETA THETA = D AND THETA = PI AND THE OTHER THETA = D AND THETA = PI THETA = D AND THETA	BELOW)	
C THETA = B = PI. C = 9 IF THE SOLUTION IS UNSPECIFIED AT C = B = PI. C	C	AND THE SOLUTION IS
C = 9 IF THE SOLUTION IS UNSPECIFIED AT C THETA = A = 0 AND THETA C NOTE 1: C NOTE 1: C IF A = 0, DO NOT USE MBDCND = 5, 6, OR 9. C NOTE 2: C NOTE 2: C NOTE 2: C IF B = PI, DO NOT USE MBDCND = 7, 8, OR 9. C NOTE 3: C NOTE 3: C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C MOBINATIONS C DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C SPECIFICATION OF THE C SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	UNSPECIFIED AT	
C = 9 IF THE SOLUTION IS UNSPECIFIED AT C THETA = A = 0 AND THETA C NOTE 1: C IF A = 0, DO NOT USE MBDCND = 5, 6, OR 9. C NOTE 2: C NOTE 2: C NOTE 2: C IF B = PI, DO NOT USE MBDCND = 7, 8, OR 9. C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C SPECIFICATION OF THE C SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	THETA = $B = PI$.
UNSPECIFIED AT C B = PI. C C C NOTE 1: IF A = 0, DO NOT USE MBDCND = 5, 6, OR 9. C C NOTE 2: IF B = PI, DO NOT USE MBDCND = 5, MOTE 2: IF B = PI, DO NOT USE MBDCND = 7, 8, OR 9. C C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: WHEN THE SOLUTION SARE COMBINATIONS C C C C C C C C C C C C C	C	
UNSPECIFIED AT C B = PI. C C C NOTE 1: IF A = 0, DO NOT USE MBDCND = 5, 6, OR 9. C C NOTE 2: IF B = PI, DO NOT USE MBDCND = 5, MOTE 2: IF B = PI, DO NOT USE MBDCND = 7, 8, OR 9. C C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: WHEN THE SOLUTION SARE COMBINATIONS C C C C C C C C C C C C C	С	= 9 IF THE SOLUTION IS
C THETA = A = 0 AND THETA = B = PI. C NOTE 1: IF A = 0, DO NOT USE MBDCND = 3, 4, OR 8, BUT INSTEAD USE MBDCND = 5, C NOTE 2: IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT THETA = 0 AND/OR THETA = PI AND THE OTHER COMBINATIONS C		
= B = PI. C C C C C C C C C C C C C C C C C C		THETA = $A = 0$ AND THETA
C C C C C C C C C C C C S S S S S S S S		
C NOTE 1: IF A = 0, DO NOT USE MBDCND = 3, 4, OR 8, C BUT INSTEAD USE MBDCND = 5, 6, OR 9. C NOTE 2: IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C PERIODICITY A SINGULAR SYSTEM RESULTS. C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
IF A = 0, DO NOT USE MBDCND = 3, 4, OR 8, C BUT INSTEAD USE MBDCND = 5, 6, OR 9. C NOTE 2: C NOTE 2: C IF B = PI, DO NOT USE MBDCND = 7, 8, OR 9. C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		NOTE 1.
3, 4, OR 8, C BUT INSTEAD USE MBDCND = 5, 6, OR 9. C C NOTE 2: IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C DERIVATIVE, OR C RESULTS. C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C SPECIFICATION OF THE C SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
C BUT INSTEAD USE MBDCND = 5, 6, OR 9. C NOTE 2: IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		IF A = 0, DO NOT USE MEDICAL =
C NOTE 2: C IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
C C C C C C C C C C C C C C C C C C C		BUT INSTEAD USE MBDCND = 5,
C NOTE 2: C IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE C SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	· •	
C IF B = PI, DO NOT USE MBDCND = 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
= 2, 3, OR 6, C BUT INSTEAD USE MBDCND = 7, 8, OR 9. C C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
BUT INSTEAD USE MBDCND = 7, 8, OR 9. C C C C C C C C C C C C C C C C C C C		IF B = PI, DO NOT USE MBDCND
8, OR 9. C C C C NOTE 3: WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C SPECIFICATION OF THE C SPECIFICATION OF THE C OR THETA = PI.	= 2, 3, OR 6,	
C C C C C C C C C C C C C C C C C C C	C	BUT INSTEAD USE MBDCND = 7 ,
C NOTE 3: C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	8, OR 9.	
C WHEN THE SOLUTION IS SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	
SPECIFIED AT C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	NOTE 3:
C THETA = 0 AND/OR THETA = PI AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	WHEN THE SOLUTION IS
AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	SPECIFIED AT	
AND THE OTHER C BOUNDARY CONDITIONS ARE COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	THETA = 0 AND/OR THETA = PI
COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	AND THE OTHER	
COMBINATIONS C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	C	BOUNDARY CONDITIONS ARE
C OF UNSPECIFIED, NORMAL DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		
DERIVATIVE, OR C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		OF IINSPECIFIED, NORMAL
C PERIODICITY A SINGULAR SYSTEM RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		or onordering, north
RESULTS. C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.	· ·	PERIODICITY A SINCHIAR SYSTEM
C THE UNIQUE SOLUTION IS DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		TERTODICITI A SINOCEAR SISIEM
DETERMINED BY C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		THE INTOIT COLUTION TO
C EXTRAPOLATION TO THE SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		11117 0141 0011 011 1014 119
SPECIFICATION OF THE C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		EVEDADOLATION TO THE
C SOLUTION AT EITHER THETA = 0 OR THETA = PI.		EVILVALOPATION IO IUE
OR THETA = PI.		COLUMNOM AM DIMIND MINERA
		SOLUTION AT EITHER THETA = 0
BUT IN THESE CASES THE RIGHT		DIE THE THE CO. C.
	C	BUT IN THESE CASES THE RIGHT

```
SIDE OF THE
                            SYSTEM WILL BE PERTURBED BY
THE CONSTANT
                           PERTRB.
C
С
С
                         BDA
C
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                           SPECIFIES THE BOUNDARY VALUES
(IF ANY) OF
                           THE SOLUTION AT THETA = A.
C
С
                           WHEN MBDCND = 1, 2, OR 7,
С
                             BDA(J) = U(A, PHI(J)),
J=1,2,...,N.
C
С
                           WHEN MBDCND = 3, 4, OR 8,
С
                             BDA(J) =
(D/DTHETA)U(A,PHI(J)),
                              J=1,2,...,N.
C
С
C
                           WHEN MBDCND HAS ANY OTHER
VALUE,
                           BDA IS A DUMMY VARIABLE.
C
С
С
                         BDB
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
С
                            SOLUTION AT THETA = B.
С
С
                           WHEN MBDCND = 1,4, OR 5,
                             BDB(J) = U(B, PHI(J)),
J=1,2,...,N.
С
                            WHEN MBDCND = 2,3, OR 6,
С
С
                             BDB(J) =
(D/DTHETA)U(B,PHI(J)),
                             J=1,2,...,N.
C
C
С
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDB IS
                           A DUMMY VARIABLE.
```

```
С
                         C,D
С
                           THE RANGE OF PHI (LONGITUDE),
С
                           I.E. C .LE. PHI .LE. D.
                           C MUST BE LESS THAN D. IF D-
С
C = 2*PI
                           PERIODIC BOUNDARY CONDITIONS
ARE USUALLY
С
                           USUALLY PRESCRIBED.
С
                         N
С
C
                           THE NUMBER OF UNKNOWNS IN THE
INTERVAL
                           (C,D). THE UNKNOWNS IN THE
PHI-DIRECTION
                           ARE GIVEN BY PHI(J) = C + (J -
0.5) DPHI,
                           J=1,2,...,N, WHERE DPHI = (D-
С
C)/N.
                           N MUST BE GREATER THAN 2.
С
С
С
                        NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                           AT PHI = C AND PHI = D.
С
                           = 0 IF THE SOLUTION IS
PERIODIC IN PHI,
                                I.E. U(I,J) = U(I,N+J).
С
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
                                PHI = C AND PHI = D
C
С
                                (SEE NOTE BELOW).
С
                           = 2 IF THE SOLUTION IS
SPECIFIED AT
                                PHI = C AND THE
DERIVATIVE OF THE
                                SOLUTION WITH RESPECT TO
PHI IS
С
                                SPECIFIED AT PHI = D
С
                                 (SEE NOTE BELOW).
```

```
= 3 IF THE DERIVATIVE OF THE
SOLUTION
                                 WITH RESPECT TO PHI IS
SPECIFIED
                                 AT PHI = C AND PHI = D.
С
C
                            = 4 IF THE DERIVATIVE OF THE
SOLUTION
                                 WITH RESPECT TO PHI IS
SPECIFIED
                                 AT PHI = C AND THE
SOLUTION IS
                                 SPECIFIED AT PHI = D
С
                                 (SEE NOTE BELOW).
С
С
                            NOTE:
                            WHEN NBDCND = 1, 2, OR 4, DO
NOT USE
                            MBDCND = 5, 6, 7, 8, OR 9
C
                            (THE FORMER INDICATES THAT
THE SOLUTION
                            IS SPECIFIED AT A POLE; THE
LATTER
                            INDICATES THE SOLUTION IS
UNSPECIFIED).
                            USE INSTEAD MBDCND = 1 \text{ OR } 2.
C
С
С
                          BDC
                            A ONE DIMENSIONAL ARRAY OF
LENGTH M THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
                            SOLUTION AT PHI = C.
C
С
С
                            WHEN NBDCND = 1 \text{ OR } 2,
                              BDC(I) = U(THETA(I), C),
С
I=1,2,...,M.
C
С
                            WHEN NBDCND = 3 OR 4,
                              BDC(I) =
(D/DPHI)U(THETA(I),C),
                              I=1,2,...,M.
C
С
                            WHEN NBDCND = 0, BDC IS A
```

```
DUMMY VARIABLE.
С
C
                          BDD
C
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH M THAT
                            SPECIFIES THE BOUNDARY VALUES
OF THE
                            SOLUTION AT PHI = D.
С
C
С
                            WHEN NBDCND = 1 OR 4,
                              BDD(I) = U(THETA(I), D),
С
I=1,2,...,M.
С
                            WHEN NBDCND = 2 \text{ OR } 3,
С
                              BDD(I) =
(D/DPHI)U(THETA(I),D),
                              I=1,2,...,M.
С
                            WHEN NBDCND = 0, BDD IS A
DUMMY VARIABLE.
С
С
                         ELMBDA
C
                            THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                            EQUATION. IF LAMBDA IS
GREATER THAN 0,
                           A SOLUTION MAY NOT EXIST.
HOWEVER,
                            HSTSSP WILL ATTEMPT TO FIND A
SOLUTION.
С
С
                          F
С
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE RIGHT SIDE
OF THE
С
                            HELMHOLTZ EOUATION.
                            FOR I=1,2,...,M AND
C
J=1,2,...,N
C
C
                             F(I,J) = F(THETA(I), PHI(J))
\mathsf{C}
                            F MUST BE DIMENSIONED AT
```

LEAST M X N.	
C	
C	IDIMF
C	THE ROW (OR FIRST) DIMENSION
OF THE ARRAY	
C	F AS IT APPEARS IN THE
PROGRAM CALLING	
C	HSTSSP. THIS PARAMETER IS
USED TO SPECIFY	
C	THE VARIABLE DIMENSION OF F.
C	IDIMF MUST BE AT LEAST M.
С	
C	
C ON OUTPUT	F
C	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	(THETA(I),PHI(J)) FOR
C	I=1,2,,M, J=1,2,,N.
C	
C	PERTRB
C	IF A COMBINATION OF PERIODIC,
DERIVATIVE,	
C	OR UNSPECIFIED BOUNDARY
CONDITIONS IS	
С	SPECIFIED FOR A POISSON
EQUATION	(
C	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
C	PERTRB IS A CONSTANT,
CALCULATED AND	
C	SUBTRACTED FROM F, WHICH
ENSURES THAT A	COLUMNON DATAMA HAMAAA MIIDN
С	SOLUTION EXISTS. HSTSSP THEN
COMPUTES	THE COLUMN DULL TO A
C LEAGE COLLADES	THIS SOLUTION, WHICH IS A
LEAST SQUARES C	SOLUTION TO THE ORIGINAL
APPROXIMATION.	SOTUTION TO THE OXIGINAT
C C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	THE SOLUTION ETOS WILL
C C	A SOLUTION; HENCE, THE
SOLUTION IS NOT	A SOLICITON, HENCE, THE
POTIOTION IN INOI	

C	UNIQUE. THE VALUE OF PERTRB
SHOULD BE	
С	SMALL COMPARED TO THE RIGHT
SIDE F.	
С	OTHERWISE, A SOLUTION IS
OBTAINED TO AN	
C	ESSENTIALLY DIFFERENT
PROBLEM.	
C	THIS COMPARISON SHOULD ALWAYS
BE MADE TO	
C	INSURE THAT A MEANINGFUL
SOLUTION HAS BEEN	
C	OBTAINED.
С	
С	IERROR
С	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
С	PARAMETERS. EXCEPT TO NUMBERS
0 AND 14,	
С	A SOLUTION IS NOT ATTEMPTED.
С	
С	= 0 NO ERROR
С	
С	= 1 A .LT. 0 OR B .GT. PI
С	
С	= 2 A .GE. B
С	
С	= 3 MBDCND .LT. 1 OR MBDCND
.GT. 9	
С	
С	= 4 C .GE. D
С	
С	= 5 N .LE. 2
С	
С	= 6 NBDCND .LT. 0 OR NBDCND
.GT. 4	
С	
С	= 7 A .GT. 0 AND MBDCND =
5, 6, OR 9	
С	
С	= 8 A $=$ 0 AND MBDCND $=$ 3,
4, OR 8	
С	
С	= 9 B .LT. PI AND MBDCND

```
.GE. 7
С
C
                           = 10 B = PI AND MBDCND =
2,3, OR 6
                           = 11 MBDCND .GE. 5 AND
NDBCND = 1, 2, OR 4
С
                           = 12 IDIMF .LT. M
С
С
                           = 13 M .LE. 2
С
С
                           = 14 LAMBDA .GT. 0
С
С
                           = 20 If the dynamic
allocation of real and
                                complex work space
required for solution
                                fails (for example if
N,M are too large
                                for your computer)
С
С
                           SINCE THIS IS THE ONLY MEANS
OF INDICATING
                           A POSSIBLY INCORRECT CALL TO
HSTSSP, THE
                           USER SHOULD TEST IERROR AFTER
THE CALL.
C I/O
                        NONE
C
C PRECISION
                       SINGLE
С
C REQUIRED FILES
fish.f, comf.f, genbun.f, gnbnaux.f, poistg.f
С
C LANGUAGE
                        FORTRAN 90
С
C HISTORY
                      WRITTEN BY ROLAND SWEET AT NCAR
IN 1977.
                         RELEASED ON NCAR'S PUBLIC
SOFTWARE LIBRARIES
С
                         IN JANUARY 1980.
                         Revised in June 2004 by John
```

Adams using	
C	Fortran 90 dynamically
allocated work space.	
C PORTABILITY	FORTRAN 90.
C ALGORITHM FINITE-	THIS SUBROUTINE DEFINES THE
C INCORPORATES BOUNDARY	DIFFERENCE EQUATIONS,
С	DATA, ADJUSTS THE RIGHT SIDE
WHEN THE SYSTEM C POISTG OR GENBUN	IS SINGULAR AND CALLS EITHER
C OF EQUATIONS.	WHICH SOLVES THE LINEAR SYSTEM
C C TIMING	FOR LARGE M AND N, THE
OPERATION COUNT	IS ROUGHLY PROPORTIONAL TO
M*N*LOG2(N).	
C ACCURACY RESULTS IN	THE SOLUTION PROCESS EMPLOYED
C SIGNIFICANT	A LOSS OF NO MORE THAN FOUR
C 64.	DIGITS FOR N AND M AS LARGE AS
C ACCURACY	MORE DETAILED INFORMATION ABOUT
C DOCUMENTATION FOR	CAN BE FOUND IN THE
C ROUTINE THAT	ROUTINE POISTG WHICH IS THE
C DIFFERENCE	ACTUALLY SOLVES THE FINITE
C C	EQUATIONS.
C REFERENCES DIRECT METHOD	U. SCHUMANN AND R. SWEET,"A
C EQUATION WITH	FOR THE SOLUTION OF POISSON'S
C A STAGGERED	NEUMANN BOUNDARY CONDITIONS ON
A DINGGRIAD	

HW3CRT

```
С
С
      file hw3crt.txt (documentation for the FISHPACK
solver HW3CRT)
С
С
С
                         copyright (c) 2005 by UCAR
С
C
              University Corporation for Atmospheric
Research
C
С
                             all rights reserved
C
                           FISHPACK90 version 1.1
С
С
С
                        A Package of Fortran 77 and 90
*
С
*
С
                       Subroutines and Example Programs
```

```
for Modeling Geophysical Processes
С
С
С
                                        by
С
С
                John Adams, Paul Swarztrauber and Roland
Sweet
C
С
                                        of
С
                  the National Center for Atmospheric
Research
С
С
                         Boulder, Colorado (80307)
U.S.A.
С
С
                             which is sponsored by
С
С
                       the National Science Foundation
С
С
С
      SUBROUTINE HW3CRT
(XS, XF, L, LBDCND, BDXS, BDXF, YS, YF, M, MBDCND, BDYS,
BDYF, ZS, ZF, N, NBDCND, BDZS, BDZF, ELMBDA, LDIMF,
С
                            MDIMF, F, PERTRB, IERROR)
\mathsf{C}
С
```

```
C DIMENSION OF
                          BDXS (MDIMF, N+1),
BDXF (MDIMF, N+1),
C ARGUMENTS
                          BDYS (LDIMF, N+1),
BDYF (LDIMF, N+1),
                          BDZS (LDIMF, M+1),
BDZF(LDIMF, M+1),
                          F(LDIMF, MDIMF, N+1)
C
C LATEST REVISION
                         June 2004
C PURPOSE
                         SOLVES THE STANDARD FIVE-POINT
FINITE
                         DIFFERENCE APPROXIMATION TO THE
HELMHOLTZ
                          EQUATION IN CARTESIAN
COORDINATES. THIS
C
                          EQUATION IS
С
С
                             (D/DX)(DU/DX) + (D/DY)(DU/DY)
+
С
                             (D/DZ)(DU/DZ) + LAMBDA*U =
F(X,Y,Z).
C
C USAGE
                          CALL HW3CRT
(XS, XF, L, LBDCND, BDXS, BDXF, YS, YF, M,
MBDCND, BDYS, BDYF, ZS, ZF, N, NBDCND,
BDZS, BDZF, ELMBDA, LDIMF, MDIMF, F,
C
                                        PERTRB, IERROR)
C
C ARGUMENTS
С
C ON INPUT
                         XS,XF
С
С
                            THE RANGE OF X, I.E. XS .LE.
X .LE. XF .
С
                            XS MUST BE LESS THAN XF.
С
С
                          L
С
                            THE NUMBER OF PANELS INTO
WHICH THE
                            INTERVAL (XS, XF) IS
SUBDIVIDED.
```

C IN THE X-DIRECTION GIVEN BY X(I) = XS+(I-1) DX FOR I=1,2,,L+1, C WHERE DX = (XF-XS)/L IS THE PANEL WIDTH. C L MUST BE AT LEAST 5. C LBDCND INDICATES THE TYPE OF BOUNDARY CONDITIONS C AT X = XS AND X = XF. C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. E = 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C SOLUTION C SPECIFIED AT C SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E = 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT	С	HENCE, THERE WILL BE L+1 GRID
I=1,2,,L+1, C WHERE DX = (XF-XS)/L IS THE PANEL WIDTH. C L MUST BE AT LEAST 5. C C LBDCND C INDICATES THE TYPE OF BOUNDARY CONDITIONS C AT X = XS AND X = XF. C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. C = 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT	POINTS C	IN THE X-DIRECTION GIVEN BY
C WHERE DX = (XF-XS)/L IS THE PANEL WIDTH. C L MUST BE AT LEAST 5. C LBDCND C INDICATES THE TYPE OF BOUNDARY CONDITIONS C AT X = XS AND X = XF. C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. E 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C X = XS AND THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT X = XF. E 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT WITH RESPECT TO X IS		X(I) = XS+(I-1)DX FOR
C LBDCND C LBDCND C INDICATES THE TYPE OF BOUNDARY CONDITIONS C AT X = XS AND X = XF. C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. C = 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT		WHERE DX = $(XF-XS)/L$ IS THE
C LBDCND C INDICATES THE TYPE OF BOUNDARY CONDITIONS C AT X = XS AND X = XF. C C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. E 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C X = XS AND THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT C WITH RESPECT TO X IS SPECIFIED AT		T. MIIST BE AT LEAST 5
C BOUNDARY CONDITIONS C AT X = XS AND X = XF. C = 0 IF THE SOLUTION IS PERIODIC IN X, C I.E. U(L+I,J,K) = U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C X = XS AND X = XF. E 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE C X = XS AND THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT X = XF. E 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT	С	
BOUNDARY CONDITIONS C C C C C C C C C C C C C		
C	BOUNDARY CONDITIONS	
C = 0 IF THE SOLUTION IS PERIODIC IN X, C		AT $X = XS$ AND $X = XF$.
C U(I,J,K). C = 1 IF THE SOLUTION IS SPECIFIED AT C		= 0 IF THE SOLUTION IS
C = 1 IF THE SOLUTION IS SPECIFIED AT C	·	I.E. U(L+I,J,K) =
SPECIFIED AT C X = XS AND X = XF. E 2 IF THE SOLUTION IS SPECIFIED AT C X = XS AND THE X = XS AND THE SOLUTION WITH RESPECT TO X IS C SPECIFIED AT X = XF. E 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. E 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT WITH RESPECT TO X IS SPECIFIED AT		= 1 TE THE SOLUTION IS
C = 2 IF THE SOLUTION IS SPECIFIED AT C		I II IIII BOHOTION 10
SPECIFIED AT C		
DERIVATIVE OF THE C SOLUTION WITH RESPECT TO X IS C SPECIFIED AT X = XF. C = 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT	SPECIFIED AT	
X IS C SPECIFIED AT X = XF. C = 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT		X = XS AND THE
C SPECIFIED AT X = XF. C = 3 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT		SOLUTION WITH RESPECT TO
SOLUTION C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT		SPECIFIED AT $X = XF$.
C WITH RESPECT TO X IS SPECIFIED AT C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT		= 3 IF THE DERIVATIVE OF THE
C X = XS AND X = XF. C = 4 IF THE DERIVATIVE OF THE SOLUTION C WITH RESPECT TO X IS SPECIFIED AT	С	WITH RESPECT TO X IS
SOLUTION C WITH RESPECT TO X IS SPECIFIED AT		X = XS AND X = XF.
C WITH RESPECT TO X IS SPECIFIED AT		= 4 IF THE DERIVATIVE OF THE
		WITH RESPECT TO X IS
X = XS AND THE SOLUTION	SPECIFIED AT	Y - YS AND THE SOLUTION
IS SPECIFIED		A - AS AND THE SOLUTION
C AT X=XF.		AT X=XF.
C BDXS	С	
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE		A TWO-DIMENSIONAL ARRAY THAT

```
VALUES OF THE DERIVATIVE OF
THE SOLUTION
C
                            WITH RESPECT TO X AT X = XS.
С
С
                            WHEN LBDCND = 3 OR 4,
С
                              BDXS(J,K) =
(D/DX)U(XS,Y(J),Z(K)),
                             J=1,2,...,M+1,
K=1,2,...,N+1.
С
                            WHEN LBDCND HAS ANY OTHER
VALUE, BDXS
                           IS A DUMMY VARIABLE. BDXS
MUST BE
                            DIMENSIONED AT LEAST
(M+1) * (N+1).
С
                         BDXF
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES THE
                            VALUES OF THE DERIVATIVE OF
THE SOLUTION
                            WITH RESPECT TO X AT X = XF.
С
С
                            WHEN LBDCND = 2 \text{ OR } 3,
С
С
                             BDXF(J,K) =
(D/DX)U(XF,Y(J),Z(K)),
                              J=1,2,...,M+1,
K=1,2,...,N+1.
                           WHEN LBDCND HAS ANY OTHER
С
VALUE, BDXF IS
                            A DUMMY VARIABLE. BDXF MUST
C
ΒE
                            DIMENSIONED AT LEAST
(M+1) * (N+1).
С
С
                          YS, YF
C
                            THE RANGE OF Y, I.E. YS .LE.
Y .LE. YF.
С
                           YS MUST BE LESS THAN YF.
С
```

C	M
С	THE NUMBER OF PANELS INTO
WHICH THE	
C	INTERVAL (YS,YF) IS
SUBDIVIDED.	
C	HENCE, THERE WILL BE M+1 GRID
POINTS IN	
C	THE Y-DIRECTION GIVEN BY Y(J)
= YS + (J-1) DY	
C	FOR $J=1,2,,M+1,$
C	WHERE DY = $(YF-YS)/M$ IS THE
PANEL WIDTH.	
C	M MUST BE AT LEAST 5.
C	
C	MBDCND
C	INDICATES THE TYPE OF
BOUNDARY CONDITIONS	
C	AT $Y = YS$ AND $Y = YF$.
C	_
C	= 0 IF THE SOLUTION IS
PERIODIC IN Y, I.E.	
C	U(I,M+J,K) = U(I,J,K).
C	= 1 IF THE SOLUTION IS
SPECIFIED AT	
C	Y = YS AND Y = YF.
C	= 2 IF THE SOLUTION IS
SPECIFIED AT	V VO 331D EUR
C	Y = YS AND THE
DERIVATIVE OF THE	COLUMNO MITTIL DECENTED TO
C	SOLUTION WITH RESPECT TO
Y IS	
C	SPECIFIED AT Y = YF.
C	= 3 IF THE DERIVATIVE OF THE
SOLUTION	TITELL DECREES TO 11 TO
C	WITH RESPECT TO Y IS
SPECIFIED AT	
C	Y = YS AND Y = YF.
C	= 4 IF THE DERIVATIVE OF THE
SOLUTION	MITTIL DEGREE TO V. TO
C C C C C C C C C C C C C C C C C C C	WITH RESPECT TO Y IS
SPECIFIED AT	
C	AT Y = YS AND THE
SOLUTION IS	apparent as
C	SPECIFIED AT Y=YF.

```
C
С
                          BDYS
C
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE DERIVATIVE
OF THE
                            SOLUTION WITH RESPECT TO Y AT
Y = YS.
С
С
                            WHEN MBDCND = 3 OR 4,
С
C
                              BDYS(I,K) =
(D/DY)U(X(I),YS,Z(K)),
                             I=1,2,...,L+1,
K=1,2,...,N+1.
C
                            WHEN MBDCND HAS ANY OTHER
VALUE, BDYS
                            IS A DUMMY VARIABLE. BDYS
MUST BE
                            DIMENSIONED AT LEAST
(L+1) * (N+1).
C
С
                          BDYF
С
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE DERIVATIVE
OF THE
                            SOLUTION WITH RESPECT TO Y AT
Y = YF.
С
С
                            WHEN MBDCND = 2 \text{ OR } 3,
С
                              BDYF(I,K) =
(D/DY)U(X(I),YF,Z(K)),
                              I=1,2,...,L+1,
K=1,2,...,N+1.
C
С
                            WHEN MBDCND HAS ANY OTHER
VALUE, BDYF
                            IS A DUMMY VARIABLE. BDYF
MUST BE
                            DIMENSIONED AT LEAST
(L+1) * (N+1).
```

```
C
С
                         ZS,ZF
C
                           THE RANGE OF Z, I.E. ZS .LE.
Z .LE. ZF.
С
                           ZS MUST BE LESS THAN ZF.
С
С
                         N
                           THE NUMBER OF PANELS INTO
WHICH THE
                           INTERVAL (ZS, ZF) IS
SUBDIVIDED.
                           HENCE, THERE WILL BE N+1 GRID
POINTS
С
                           IN THE Z-DIRECTION GIVEN BY
С
                           Z(K) = ZS+(K-1)DZ FOR
K=1,2,...,N+1,
                           WHERE DZ = (ZF-ZS)/N IS THE
PANEL WIDTH.
                           N MUST BE AT LEAST 5.
C
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                           AT Z = ZS AND Z = ZF.
С
С
                           = 0 IF THE SOLUTION IS
С
PERIODIC IN Z, I.E.
C
                                U(I,J,N+K) = U(I,J,K).
                           = 1 IF THE SOLUTION IS
С
SPECIFIED AT
C
                                Z = ZS AND Z = ZF.
C
                           = 2 IF THE SOLUTION IS
SPECIFIED AT
                                Z = ZS AND THE
DERIVATIVE OF THE
C
                                SOLUTION WITH RESPECT TO
Z IS
                                SPECIFIED AT Z = ZF.
C
                           = 3 IF THE DERIVATIVE OF THE
С
SOLUTION
                                WITH RESPECT TO Z IS
SPECIFIED AT
C
                                Z = ZS AND Z = ZF.
                           = 4 IF THE DERIVATIVE OF THE
```

```
SOLUTION
                                 WITH RESPECT TO Z IS
SPECIFIED AT
                                 Z = ZS AND THE SOLUTION
IS SPECIFIED
                                 AT Z=ZF.
С
С
                         BDZS
С
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE DERIVATIVE
OF THE
                            SOLUTION WITH RESPECT TO Z AT
С
Z = ZS.
С
                            WHEN NBDCND = 3 OR 4,
С
С
                             BDZS(I,J) =
(D/DZ)U(X(I),Y(J),ZS),
                              I=1,2,...,L+1,
J=1,2,...,M+1.
                            WHEN NBDCND HAS ANY OTHER
C
VALUE, BDZS
                            IS A DUMMY VARIABLE. BDZS
MUST BE
                            DIMENSIONED AT LEAST
(L+1)*(M+1).
С
                          BDZF
C.
                            A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES
                            THE VALUES OF THE DERIVATIVE
С
OF THE
                            SOLUTION WITH RESPECT TO Z AT
Z = ZF.
С
С
                            WHEN NBDCND = 2 \text{ OR } 3,
С
                             BDZF(I,J) =
(D/DZ)U(X(I),Y(J),ZF),
                             I=1,2,...,L+1,
J=1,2,...,M+1.
```

C	WHEN NBDCND HAS ANY OTHER
VALUE, BDZF C	IS A DUMMY VARIABLE. BDZF
MUST BE	IS A DOMMI VARIABLE. BDZF
C	DIMENSIONED AT LEAST
(L+1) * (M+1).	
C	
C	ELMBDA THE CONSTANT LAMBDA IN THE
HELMHOLTZ	THE CONSTANT DAMBDA IN THE
C	EQUATION. IF LAMBDA .GT. 0, A
SOLUTION	
С	MAY NOT EXIST. HOWEVER,
HW3CRT WILL	ADDRADD DO ETNID A COLUMNAL
C	ATTEMPT TO FIND A SOLUTION.
C	LDIMF
С	THE ROW (OR FIRST) DIMENSION
OF THE	
C	ARRAYS F, BDYS, BDYF, BDZS, AND
BDZF AS IT	APPEARS IN THE PROGRAM
CALLING HW3CRT.	ALLEANS IN THE TROGRAM
С	THIS PARAMETER IS USED TO
SPECIFY THE	
C	VARIABLE DIMENSION OF THESE
ARRAYS.	LDIMF MUST BE AT LEAST L+1.
C	DIME MOST DE AT DEAST DIT.
С	MDIMF
С	THE COLUMN (OR SECOND)
DIMENSION OF THE	
C FIRST) DIMENSION	ARRAY F AND THE ROW (OR
C DIMENSION	OF THE ARRAYS BDXS AND BDXF
AS IT APPEARS	01 111 111110 1110 1111 1111
C	IN THE PROGRAM CALLING
HW3CRT. THIS	
C THE VARIABLE	PARAMETER IS USED TO SPECIFY
C C	DIMENSION OF THESE ARRAYS.
C	MDIMF MUST BE AT LEAST M+1.
С	
С	F

C DIMENSION ASS	A THREE-	-DIMENSIONAL ARRAY OF	
DIMENSION AT C	AT LEAST (L+1) * (M+1) * (N+1),		
SPECIFYING THE	VALUES (OF THE RIGHT SIDE OF	
THE HELMHOLZ C	EQUATION AND BOUNDARY VALUES		
(IF ANY).			
C AS FOLLOWS:	ON THE INTERIOR, F IS DEFINED		
C J=2,3,,M,	FOR I=2,3,,L,		
C C	AND K=2, F(I,J,K)	3,, N = $F(X(I), Y(J), Z(K))$.	
C C	ON THE E	BOUNDARIES, F IS	
DEFINED AS FOLLOWS:	FOR J=1,	2,,M+1,	
K=1,2,,N+1,	AND I=1,2,,L+1		
C	LBDCND	F(1,J,K)	
F(L+1,J,K)			
 C			
C F(XS,Y(J),Z(K))	0	F(XS,Y(J),Z(K))	
C U(XF, Y(J), Z(K))	1	U(XS,Y(J),Z(K))	
С	2	U(XS,Y(J),Z(K))	
F(XF,Y(J),Z(K)) C	3	F(XS,Y(J),Z(K))	
F(XF,Y(J),Z(K)) C	4	F(XS,Y(J),Z(K))	
U(XF,Y(J),Z(K)) C			
C F(I,M+1,K)	MBDCND	F(I,1,K)	
C			
C	0	F(X(I),YS,Z(K))	
	U		

F(X(I), YS, Z(K))	1	U(X(I),YS,Z(K))
U(X(I),YF,Z(K))	<u> </u>	0 (21(1) / 10/21(11))
C F(X(I),YF,Z(K))	2	U(X(I),YS,Z(K))
C (X(1), 1F, Z(N))	3	F(X(I),YS,Z(K))
F(X(I),YF,Z(K))	4	E(V/T) VC 7/V)
U(X(I),YF,Z(K))	4	F(X(I),YS,Z(K))
C	NIDDCNID	E/T T 1\
F(I,J,N+1)	NBDCND	F(I,J,1)
С		
C		
C F(X(I),Y(J),ZS)	0	F(X(I),Y(J),ZS)
C (X(1),1(0),25)	1	U(X(I),Y(J),ZS)
U(X(I),Y(J),ZF)	2	U(X(I),Y(J),ZS)
F(X(I),Y(J),ZF)	۷	0 (2(1),1(0),25)
C F(X(I),Y(J),ZF)	3	F(X(I),Y(J),ZS)
C (X(1),1(0),21)	4	F(X(I),Y(J),ZS)
U(X(I),Y(J),ZF) C		
C	NOTE:	
C THE SOLUTION	IF THE 7	TABLE CALLS FOR BOTH
C	U AND TH	HE RIGHT SIDE F ON A
BOUNDARY,	THEN THE	E SOLUTION MUST BE
SPECIFIED.		
C		
	F	
C U(I,J,K) OF THE	CONTAINS	S THE SOLUTION
C C	FINITE I	DIFFERENCE
APPROXIMATION FOR THE	CRID POI	INT (X(I),Y(J),Z(K))
FOR		
C		,L+1, J=1,2,,M+1,
C	TIND IV-I	, ∠, , IN I ⊥ .

C	DEDUID
C	PERTRB
C	IF A COMBINATION OF PERIODIC
OR DERIVATIVE	
C	BOUNDARY CONDITIONS IS
SPECIFIED FOR A	
C C	DOTCCON FOUNTION (IAMBDA -
	POISSON EQUATION (LAMBDA =
0), A SOLUTION	
C	MAY NOT EXIST. PERTRB IS A
CONSTANT,	
С	CALCULATED AND SUBTRACTED
FROM F, WHICH	
C Willell	ENGIDEC MIAM A COLUMION
	ENSURES THAT A SOLUTION
EXISTS. PWSCRT	
C	THEN COMPUTES THIS SOLUTION,
WHICH IS A	
C	LEAST SQUARES SOLUTION TO THE
ORIGINAL	
C	APPROXIMATION. THIS SOLUTION
	APPROXIMATION. THIS SOLUTION
IS NOT	
C	UNIQUE AND IS UNNORMALIZED.
THE VALUE OF	
C	PERTRB SHOULD BE SMALL
COMPARED TO THE	
C	THE RIGHT SIDE F. OTHERWISE,
	INE RIGHT SIDE F. OTHERWISE,
A SOLUTION	
C	IS OBTAINED TO AN ESSENTIALLY
DIFFERENT	
C	PROBLEM. THIS COMPARISON
SHOULD ALWAYS	
C	BE MADE TO INSURE THAT A
MEANINGFUL	DE PADE TO INSOIRE THAT A
C	SOLUTION HAS BEEN OBTAINED.
C	
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS. EXCEPT FOR
	TANAMETENO. EACETI FOR
NUMBERS 0 AND 12,	
C	A SOLUTION IS NOT ATTEMPTED.
C	
C	= 0 NO ERROR
C	= 1 XS .GE. XF
C	= 2 L .LT. 5
	- 2 11 • 11 • J

```
= 3 LBDCND .LT. 0 .OR.
LBDCND .GT. 4
C
                          = 4 YS .GE. YF
С
                         = 5 M .LT. 5
                          = 6 MBDCND .LT. 0 .OR.
С
MBDCND .GT. 4
                          = 7 ZS .GE. ZF
C
С
                         = 8 N .LT. 5
                         = 9 NBDCND .LT. 0 .OR.
С
NBDCND .GT. 4
                          = 10 LDIMF .LT. L+1
C
                          = 11 MDIMF .LT. M+1
С
                          = 12 LAMBDA .GT. 0
                         = 20 If the dynamic
allocation of real and
                              complex work space
required for solution
                              fails (for example if
N,M are too large
                              for your computer)
С
С
                        SINCE THIS IS THE ONLY MEANS
OF INDICATING
                      A POSSIBLY INCORRECT CALL TO
HW3CRT, THE
                       USER SHOULD TEST IERROR AFTER
C
THE CALL.
C SPECIAL CONDITIONS NONE
С
C I/O
                       NONE
C
C PRECISION
                       SINGLE
C REQUIRED Files
fish.f,pois3d.f,fftpack.f,comf.f
C LANGUAGE
                       FORTRAN 90
С
C HISTORY
                      WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE
                       1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE
                  LIBRARIES IN JANUARY 1980.
```

C	Revised in June 2004 by John
Adams using	
C	Fortran 90 dynamically
allocated work space.	
C	
C PORTABILITY	FORTRAN 90
C	
C ALGORITHM	THIS SUBROUTINE DEFINES THE
FINITE DIFFERENCE	
C	EQUATIONS, INCORPORATES
	EQUATIONS, INCOMINATES
BOUNDARY DATA, AND	AD THOMO MILE DIGITE GIDE OF
C	ADJUSTS THE RIGHT SIDE OF
SINGULAR SYSTEMS AND	
C	THEN CALLS POIS3D TO SOLVE THE
SYSTEM.	
C	
C TIMING	FOR LARGE L, M AND N, THE
OPERATION COUNT	
C	IS ROUGHLY PROPORTIONAL TO
C	L*M*N*(LOG2(L)+LOG2(M)+5),
C	BUT ALSO DEPENDS ON INPUT
PARAMETERS LBDCND	
C	AND MBDCND.
C	
C ACCURACY	THE SOLUTION PROCESS EMPLOYED
RESULTS IN	
C	A LOSS OF NO MORE THAN FOUR
SIGNIFICANT	
C	DIGITS FOR L, M AND N AS LARGE
AS 32.	210110 1011 2, 11 11.2 11 11.0 = 1.00
C	MORE DETAILED INFORMATION ABOUT
ACCURACY	
C	CAN BE FOUND IN THE
DOCUMENTATION FOR	CAN DE LOOND IN THE
C C	ROUTINE POIS3D WHICH IS THE
	ROOTINE FOISSD WITCH IS THE
ROUTINE THAT	ACMIALLY COLUMN MILE EINITHE
C	ACTUALLY SOLVES THE FINITE
DIFFERENCE	DOLLAMIONO
C	EQUATIONS.
C	NOVE
C REFERENCES	NONE
C********	*********

HWSCRT

```
file hwscrt.txt (documentation for the FISHPACK
solver HWSCRT)
С
                        copyright (c) 2005 by UCAR
С
             University Corporation for Atmospheric
Research
                            all rights reserved
C
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
```

```
C
               John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                      of
С
                the National Center for Atmospheric
Research
                        Boulder, Colorado (80307)
U.S.A.
C
С
                           which is sponsored by
С
С
                      the National Science Foundation
С
С
      SUBROUTINE HWSCRT
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                          ELMBDA, F, IDIMF, PERTRB, IERROR)
С
C DIMENSION OF
                         BDA(N),
                                       BDB(N),
BDC (M), BDD (M),
C ARGUMENTS
                         F(IDIMF, N)
C LATEST REVISION
                         June 2004
C PURPOSE
                          SOLVES THE STANDARD FIVE-POINT
FINITE
С
                          DIFFERENCE APPROXIMATION TO THE
HELMHOLTZ
                          EQUATION IN CARTESIAN
```

```
COORDINATES. THIS
С
                          EQUATION IS
C
С
                            (D/DX)(DU/DX) + (D/DY)(DU/DY)
С
                            + LAMBDA*U = F(X,Y).
С
C USAGE
                          CALL HWSCRT
(A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
C
                                        PERTRB, IERROR)
C
C ARGUMENTS
C ON INPUT
                          A,B
С
                            THE RANGE OF X, I.E., A .LE.
X .LE. B.
С
                            A MUST BE LESS THAN B.
С
С
С
                            THE NUMBER OF PANELS INTO
WHICH THE
                            INTERVAL (A, B) IS SUBDIVIDED.
C
                            HENCE, THERE WILL BE M+1 GRID
POINTS
                            IN THE X-DIRECTION GIVEN BY
C
                            X(I) = A+(I-1)DX FOR I =
1, 2, \ldots, M+1,
                            WHERE DX = (B-A)/M IS THE
PANEL WIDTH.
C
                            M MUST BE GREATER THAN 3.
С
С
                          MBDCND
                            INDICATES THE TYPE OF
BOUNDARY CONDITIONS
С
                            AT X = A AND X = B.
С
                            = 0 IF THE SOLUTION IS
C
PERIODIC IN X,
C
                                  I.E., U(I,J) = U(M+I,J).
                            = 1 IF THE SOLUTION IS
SPECIFIED AT
С
                                 X = A AND X = B.
                            = 2 IF THE SOLUTION IS
```

SPECIFIED AT	
C	X = A AND THE DERIVATIVE
OF THE	
C	SOLUTION WITH RESPECT TO
X IS	
C	SPECIFIED AT $X = B$.
C	= 3 IF THE DERIVATIVE OF THE
SOLUTION	
C	WITH RESPECT TO X IS
SPECIFIED AT	WIII 1001201 10 11 10
C	AT $X = A$ AND $X = B$.
C	= 4 IF THE DERIVATIVE OF THE
	- 4 If the DERIVATIVE OF the
SOLUTION	
C	WITH RESPECT TO X IS
SPECIFIED AT	
C	X = A AND THE SOLUTION
IS SPECIFIED	
C	AT $X = B$.
С	
C	BDA
С	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1 THAT	
C	SPECIFIES THE VALUES OF THE
DERIVATIVE	
C	OF THE SOLUTION WITH RESPECT
TO X AT $X = A$.	OF THE SOLUTION WITH IMBLECT
C	LILIENI MODONIO 2 OD 4
C	WHEN MBDCND = $3 \text{ OR } 4$,
C	
С	BDA(J) = (D/DX)U(A,Y(J)), J
= 1, 2,, N+1.	
C	
C	WHEN MBDCND HAS ANY OTHER
VALUE, BDA IS	
C	A DUMMY VARIABLE.
C	
С	BDB
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1	
C	THAT SPECIFIES THE VALUES OF
THE DERIVATIVE	TIME OF COLUMN VILLOUD OF
C	OF THE SOLUTION WITH RESPECT
	OF THE SOFOTION MITH KESSECT
TO X AT $X = B$.	
C	

```
WHEN MBDCND = 2 \text{ OR } 3,
С
C
                              BDB(J) = (D/DX)U(B,Y(J)), J
= 1, 2, ..., N+1
                            WHEN MBDCND HAS ANY OTHER
C
VALUE BDB IS A
                            DUMMY VARIABLE.
C
С
                          C,D
                            THE RANGE OF Y, I.E., C .LE.
С
Y .LE. D.
                            C MUST BE LESS THAN D.
C
С
С
                         Ν
C
                            THE NUMBER OF PANELS INTO
WHICH THE
                            INTERVAL (C,D) IS SUBDIVIDED.
С
HENCE,
                            THERE WILL BE N+1 GRID POINTS
IN THE
                            Y-DIRECTION GIVEN BY Y(J) =
C+(J-1)DY
С
                            FOR J = 1, 2, ..., N+1, WHERE
                            DY = (D-C)/N IS THE PANEL
С
WIDTH.
                            N MUST BE GREATER THAN 3.
C
C
С
                          NBDCND
                            INDICATES THE TYPE OF
BOUNDARY CONDITIONS AT
                            Y = C AND Y = D.
С
                            = 0 IF THE SOLUTION IS
PERIODIC IN Y,
С
                                 I.E., U(I,J) = U(I,N+J).
                            = 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
С
                                 SPECIFIED AT Y = D.
C
                            = 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
С
SOLUTION
                                 WITH RESPECT TO Y IS
SPECIFIED AT
                                Y = C AND THE SOLUTION
IS SPECIFIED
                                AT Y = D.
C
С
С
                         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.
C
С
                            WHEN NBDCND = 3 OR 4,
С
                             BDC(I) = (D/DY)U(X(I),C), I
C
= 1, 2, ..., M+1
С
С
                           WHEN NBDCND HAS ANY OTHER
VALUE, BDC IS
C
                           A DUMMY VARIABLE.
С
С
                         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.
C
C
                            WHEN NBDCND = 2 \text{ OR } 3,
С
C
                             BDD(I) = (D/DY)U(X(I),D), I
= 1, 2, ..., M+1
```

C	WHEN NBDCND HAS AN	Y OTHER
VALUE, BDD IS		
C	A DUMMY VARIABLE.	
C	ELMBDA	
C HELMHOLTZ	THE CONSTANT LAMBD	A IN THE
C C	EQUATION. IF LAMB	DA .GT. 0,
A SOLUTION	MALL NOT THE ST	
C HWSCRT WILL	MAY NOT EXIST. HO	WEVER,
С	ATTEMPT TO FIND A	SOLUTION.
C	F	
C	A TWO-DIMENSIONAL	ARRAY, OF
DIMENSION AT	T E A C	CDECTEVING
VALUES OF THE	LEAST (M+1) * (N+1),	SPECIFIING
C	RIGHT SIDE OF THE	HELMHOLTZ
EQUATION AND C	BOUNDARY VALUES (I	F ANY).
С		·
C AS FOLLOWS:	ON THE INTERIOR, F	IS DEFINED
C C	FOR $I = 2, 3,, M$	AND J =
2,3,,N	F(I,J) = F(X(I),Y(I))	T\ \
C	$\Gamma(1,0) - \Gamma(\Lambda(1),1)$	0)).
C	ON THE BOUNDARIES,	F IS
DEFINED AS FOLLOWS:	FOR J=1,2,,N+1,	
I=1,2,,M+1,		
C	MBDCND F(1,J)	
F(M+1,J)	1220112 1 (170)	
C		
C		
C	0 F(A,Y(J))
F(A, Y(J)) C	1 U(A,Y(J	.))
U(B,Y(J))		
C	2 U(A, Y(J))

F(B,Y(J))		
C	3	F(A,Y(J))
F(B,Y(J))	4	F(A,Y(J))
U(B,Y(J))		
C		
C	NBDCND	F(I,1)
F(I,N+1)		
C	0	= (T, (T) , G)
C F(X(I),C)	0	F(X(I),C)
С	1	U(X(I),C)
U(X(I),D) C	2	U(X(I),C)
F(X(I),D)		
C F(X(I),D)	3	F(X(I),C)
C (X(1),D)	4	F(X(I),C)
U(X(I),D)		
C	NOTE:	
С		ABLE CALLS FOR BOTH
THE SOLUTION U	AND THE E	RIGHT SIDE F AT A
CORNER THEN THE		CIOIII OIDH I III II
C	SOLUTION	MUST BE SPECIFIED.
C	IDIMF	
С	THE ROW ((OR FIRST) DIMENSION
OF THE ARRAY C	F AS TT A	APPEARS IN THE
PROGRAM CALLING	1 110 11 1	
C USED TO SPECIFY	HWSCRT.	THIS PARAMETER IS
C C	THE VARIA	ABLE DIMENSION OF F.
IDIMF MUST		ACITI MILI
C	BE AT LEA	AST M+1 .
С		
C ON OUTPUT	F CONTAINS	THE SOLUTION U(I,J)
	CONTAINS	TILL DOLLOTTON O(T,O)

OF THE FINITE C C THE GRID POINT C C (X(I),Y(J)), I = 1,2,,M+1, J = 1,2,,N+1 C C PERTRB C FROM FOR A C C C FROM F, WHICH C EXISTS. HWSCRT C C CRIGINAL C C CRIGINAL C C CRIGINAL C C C CRIGINAL C C C CONSTANT, C C C C C C C C C C C C C C C C C C C		
THE GRID POINT C C C C C C C C C C C C C C C C C C C		
C (X(I),Y(J)), I = 1,2,,M+1, C J = 1,2,,M+1 C C C PERTRB C IF A COMBINATION OF PERIODIC OR DERIVATIVE C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS C DETAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR AN ERROR FLAG THAT INDICATES		DIFFERENCE APPROXIMATION FOR
C J=1,2,,N+1 . C PERTRB C IF A COMBINATION OF PERIODIC OR DERIVATIVE C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR AN ERROR FLAG THAT INDICATES		(37/T) 37/T)
C PERTRB C IF A COMBINATION OF PERIODIC OR DERIVATIVE C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS NOT UNIQUE. THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR AN ERROR FLAG THAT INDICATES		
C PERTRB C IF A COMBINATION OF PERIODIC C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR		$\cup = \bot, \angle, \ldots, \mathbb{N} + \bot .$
C IF A COMBINATION OF PERIODIC OR DERIVATIVE C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = O), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR C IERROR C IERROR AN ERROR FLAG THAT INDICATES		DEDED
OR DERIVATIVE C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS NOT UNIQUE. THE C FERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR C IERROR C AN ERROR FLAG THAT INDICATES		
C BOUNDARY CONDITIONS IS SPECIFIED FOR A C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE C SOLUTION IS NOT UNIQUE. THE C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C IERROR C IERROR AN ERROR FLAG THAT INDICATES		IF A COMBINATION OF PERIODIC
SPECIFIED FOR A C C CONSTANT, C C CALCULATED AND SUBTRACTED FROM F, WHICH C EXISTS. HWSCRT C CORIGINAL C CORIGINAL C C C C C C C C C C C C C C C C C C C		DOLINDADY CONDITIONS TO
C POISSON EQUATION (LAMBDA = 0), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR AN ERROR FLAG THAT INDICATES		BOUNDARI CONDITIONS IS
O), A SOLUTION C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR		DOTSSON FOUNTION (LAMBDA -
C MAY NOT EXIST. PERTRB IS A CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR C IERROR C IERROR C AN ERROR FLAG THAT INDICATES		FOISSON EQUATION (LANDDA -
CONSTANT, C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR C IERROR C AN ERROR FLAG THAT INDICATES	. ,	MAY NOT FYICT DEDTOR IC A
C CALCULATED AND SUBTRACTED FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C IERROR AN ERROR FLAG THAT INDICATES		MAI NOI EXIST: TEXTION IS A
FROM F, WHICH C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR C IERROR C AN ERROR FLAG THAT INDICATES		CALCULATED AND SUBTRACTED
C ENSURES THAT A SOLUTION EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C IERROR C AN ERROR FLAG THAT INDICATES		
EXISTS. HWSCRT C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRE SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	·	ENSURES THAT A SOLUTION
C THEN COMPUTES THIS SOLUTION, WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	_ ~	
WHICH IS A C LEAST SQUARES SOLUTION TO THE ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C C IERROR C IERROR AN ERROR FLAG THAT INDICATES		THEN COMPUTES THIS SOLUTION,
ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	WHICH IS A	,
ORIGINAL C APPROXIMATION. THIS SOLUTION PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	C	LEAST SQUARES SOLUTION TO THE
PLUS ANY C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C IERROR AN ERROR FLAG THAT INDICATES	ORIGINAL	~
C CONSTANT IS ALSO A SOLUTION. HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	С	APPROXIMATION. THIS SOLUTION
HENCE, THE C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	PLUS ANY	
C SOLUTION IS NOT UNIQUE. THE VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	С	CONSTANT IS ALSO A SOLUTION.
VALUE OF C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	HENCE, THE	
C PERTRB SHOULD BE SMALL COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	C	SOLUTION IS NOT UNIQUE. THE
COMPARED TO THE C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	VALUE OF	
C RIGHT SIDE F. OTHERWISE, A SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	C	PERTRB SHOULD BE SMALL
SOLUTION IS C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR AN ERROR FLAG THAT INDICATES	COMPARED TO THE	
C OBTAINED TO AN ESSENTIALLY DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	C	RIGHT SIDE F. OTHERWISE, A
DIFFERENT C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	SOLUTION IS	
C PROBLEM. THIS COMPARISON SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C IERROR C AN ERROR FLAG THAT INDICATES	С	OBTAINED TO AN ESSENTIALLY
SHOULD ALWAYS C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C IERROR C AN ERROR FLAG THAT INDICATES	DIFFERENT	
C BE MADE TO INSURE THAT A MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C C IERROR C AN ERROR FLAG THAT INDICATES	С	PROBLEM. THIS COMPARISON
MEANINGFUL C SOLUTION HAS BEEN OBTAINED. C C C IERROR C AN ERROR FLAG THAT INDICATES	SHOULD ALWAYS	
C SOLUTION HAS BEEN OBTAINED. C IERROR C AN ERROR FLAG THAT INDICATES	С	BE MADE TO INSURE THAT A
C		
C IERROR C AN ERROR FLAG THAT INDICATES		SOLUTION HAS BEEN OBTAINED.
C AN ERROR FLAG THAT INDICATES		
		AN ERROR FLAG THAT INDICATES
INVALID INPUT	INVALID INPUT	

C NUMBERS 0 AND 6,	PARAMETERS. EXCEPT FOR
С	A SOLUTION IS NOT ATTEMPTED.
C	= 0 NO ERROR
C	= 1 A .GE. B
C	= 2 MBDCND .LT. 0 OR MBDCND
.GT. 4	
.G1. 4	= 3 C .GE. D
C	= 4 N .LE. 3
C	
	= 5 NBDCND .LT. 0 OR NBDCND
.GT. 4	
C	= 6 LAMBDA .GT. 0
С	= 7 IDIMF .LT. M+1
С	= 8 M .LE. 3
C	= 20 If the dynamic
allocation of real and	
С	complex work space
required for solution	
C	fails (for example if
N,M are too large	
С	for your computer)
С	
C	SINCE THIS IS THE ONLY MEANS
OF INDICATING	
С	A POSSIBLY INCORRECT CALL TO
HWSCRT, THE	
С	USER SHOULD TEST IERROR AFTER
THE CALL.	
С	
С	
C SPECIAL CONDITIONS	NONE
C	
C I/O	NONE
C	
C PRECISION	SINGLE
C	
C REQUIRED files	
fish.f,genbun.f,gnbnaux.	f,comf.f
C	
C LANGUAGE	FORTRAN 90
С	
C HISTORY	WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE	

C PUBLIC SOFTWARE	1970'S. RELEASED ON NCAR'S
C	LIBRARIES IN JANUARY 1980.
C	Revised in June 2004 by John
Adams using	nevised in dune 2004 by doing
C C	Fortran 90 dynamically
allocated work space.	Torcian 90 dynamicarry
C C	
C PORTABILITY	FORTRAN 90
C	I OIKIIY IV 90
C ALGORITHM	THE ROUTINE DEFINES THE FINITE
DIFFERENCE	
C	EQUATIONS, INCORPORATES
BOUNDARY DATA, AND	ngorii iono, incom oranic
C	ADJUSTS THE RIGHT SIDE OF
SINGULAR SYSTEMS	
C	AND THEN CALLS GENBUN TO SOLVE
THE SYSTEM.	
C	
C TIMING	FOR LARGE M AND N, THE
OPERATION COUNT	,
C	IS ROUGHLY PROPORTIONAL TO
C	M*N* (LOG2 (N)
С	BUT ALSO DEPENDS ON INPUT
PARAMETERS NBDCND	
С	AND MBDCND.
С	
C ACCURACY	THE SOLUTION PROCESS EMPLOYED
RESULTS IN A LOSS	
С	OF NO MORE THAN THREE
SIGNIFICANT DIGITS FOR N	
C	AND M AS LARGE AS 64. MORE
DETAILS ABOUT	
С	ACCURACY CAN BE FOUND IN THE
DOCUMENTATION FOR	
С	SUBROUTINE GENBUN WHICH IS THE
ROUTINE THAT	
C	SOLVES THE FINITE DIFFERENCE
EQUATIONS.	
C	
C REFERENCES	SWARZTRAUBER, P. AND R. SWEET,
"EFFICIENT	
C	FORTRAN SUBPROGRAMS FOR THE
SOLUTION OF	

HWSCSP

```
С
     file hwscsp.txt (documentation for the FISHPACK
С
solver HWSCSP)
С
С
С
                        copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
C
С
                             all rights reserved
C
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
*
С
С
                       Subroutines and Example Programs
```

```
for Modeling Geophysical Processes
С
С
С
                                      by
С
С
                John Adams, Paul Swarztrauber and Roland
Sweet
C
С
                                      of
С
                 the National Center for Atmospheric
Research
С
С
                        Boulder, Colorado (80307)
U.S.A.
С
                           which is sponsored by
С
С
С
                      the National Science Foundation
С
С
С
      SUBROUTINE HWSCSP
(INTL, TS, TF, M, MBDCND, BDTS, BDTF, RS, RF, N, NBDCND,
BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
С
С
                          BDTS(N+1), BDTF(N+1),
C DIMENSION OF
```

```
BDRS (M+1), BDRF (M+1),
C ARGUMENTS
                          F(IDIMF, N+1)
                         June 2004
C LATEST REVISION
С
C PURPOSE
                         SOLVES A FINITE DIFFERENCE
APPROXIMATION
                          TO THE MODIFIED HELMHOLTZ
EQUATION IN
                          SPHERICAL COORDINATES ASSUMING
AXISYMMETRY
                          (NO DEPENDENCE ON LONGITUDE).
THE EQUATION
                          IS
С
С
С
                             (1/R**2) (D/DR) ((R**2) (D/DR) U)
+
С
C
(1/(R**2)SIN(THETA))(D/DTHETA)
С
С
                             (SIN(THETA)(D/DTHETA)U) +
С
С
                             (LAMBDA/(RSIN(THETA))**2)U =
F(THETA, R).
C
                          THIS TWO DIMENSIONAL MODIFIED
HELMHOLTZ
                          EQUATION RESULTS FROM THE
FOURIER TRANSFORM
                          OF THE THREE DIMENSIONAL
POISSON EQUATION.
С
C USAGE
                          CALL HWSCSP
(INTL, TS, TF, M, MBDCND, BDTS, BDTF,
RS, RF, N, NBDCND, BDRS, BDRF, ELMBDA,
F, IDIMF, PERTRB, IERROR, W)
C
C ARGUMENTS
C ON INPUT
                          INTL
C
                            = 0 ON INITIAL ENTRY TO
HWSCSP OR IF ANY
```

C	OF THE ARGUMENTS RS, RF,
N, NBDCND	
C	ARE CHANGED FROM A
PREVIOUS CALL.	
C	= 1 IF RS, RF, N, NBDCND ARE
ALL UNCHANGED	
C	FROM PREVIOUS CALL TO
	FROM FREVIOUS CALL TO
HWSCSP.	
C	
С	NOTE:
C	A CALL WITH INTL=0 TAKES
APPROXIMATELY	
C	1.5 TIMES AS MUCH TIME AS A
CALL WITH	
C	INTL = 1 . ONCE A CALL WITH
INTL = 0	TIVIL I • ONCE 71 CALLE WITH
	IIAC DEENI MADE MIIENI CIIDCECIIENIM
C	HAS BEEN MADE THEN SUBSEQUENT
SOLUTIONS	
C	CORRESPONDING TO DIFFERENT F,
BDTS, BDTF,	
C	BDRS, BDRF CAN BE OBTAINED
FASTER WITH	
С	<pre>INTL = 1 SINCE INITIALIZATION</pre>
IS NOT	
C	REPEATED.
C	
C	TS,TF
	•
C	THE RANGE OF THETA
(COLATITUDE), I.E.,	
С	TS .LE. THETA .LE. TF. TS
MUST BE LESS	
C	THAN TF. TS AND TF ARE IN
RADIANS. A TS OF	
C	ZERO CORRESPONDS TO THE NORTH
POLE AND A	
C	TF OF PI CORRESPONDS TO THE
	IF OF II COMESTONDS TO THE
SOUTH POLE.	
C	
C	**** IMPORTANT ****
C	
C	IF TF IS EQUAL TO PI THEN IT
MUST BE	
С	COMPUTED USING THE STATEMENT
C	TF = PIMACH(DUM). THIS

INSURES THAT TF	
С	IN THE USER'S PROGRAM IS
EQUAL TO PI IN	
С	THIS PROGRAM WHICH PERMITS
SEVERAL TESTS	
C	OF THE INPUT PARAMETERS THAT
OTHERWISE	
C	WOULD NOT BE POSSIBLE.
C	
C	M
C	THE NUMBER OF PANELS INTO
WHICH THE	TNIMEDIAT (MC ME) TO
C	INTERVAL (TS,TF) IS
SUBDIVIDED.	HENCE MHEDE MILL DE MILL COID
POINTS	HENCE, THERE WILL BE M+1 GRID
C	IN THE THETA-DIRECTION GIVEN
BY	IN THE HELA-DIRECTION GIVEN
C	THETA(K) = $(I-1)$ DTHETA+TS FOR
C	$I = 1, 2, \dots, M+1$, WHERE DTHETA
= (TF-TS)/M	1 1/2//11.1/ WILLE DITE
, ,	
$\mid C \mid$	IS THE PANEL WIDTH.
C	IS THE PANEL WIDTH.
CCC	IS THE PANEL WIDTH. MBDCND
C	
C	MBDCND
C C C	MBDCND
C C C BOUNDARY CONDITION	MBDCND INDICATES THE TYPE OF
C C BOUNDARY CONDITION C	MBDCND INDICATES THE TYPE OF
C C C BOUNDARY CONDITION C TF.	MBDCND INDICATES THE TYPE OF
C C C BOUNDARY CONDITION C TF. C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA =
C C C BOUNDARY CONDITION C TF. C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA =
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS
C C C BOUNDARY CONDITION C TF. C C C SPECIFIED AT C TF.	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS THETA = TS AND THETA = = 2 IF THE SOLUTION IS
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS THETA = TS AND THETA =
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C DERIVATIVE OF THE	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS THETA = TS AND THETA = = 2 IF THE SOLUTION IS THETA = TS AND THE
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C DERIVATIVE OF THE C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1 IF THE SOLUTION IS THETA = TS AND THETA = = 2 IF THE SOLUTION IS
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C TF. C TF. C TF. C TF. C THETA IS	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C TF. C SPECIFIED AT C THETA IS C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C TF. C TF. C TF. C TF. C TF. C THETA IS C C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C TF. C TF. C SPECIFIED AT C C THETA IS C C C C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1
C C C BOUNDARY CONDITION C TF. C C SPECIFIED AT C TF. C SPECIFIED AT C TF. C TF. C TF. C TF. C TF. C THETA IS C C	MBDCND INDICATES THE TYPE OF AT THETA = TS AND THETA = = 1

SPECIFIED		
C		AT THETA = TS AND THETA
= TF		
С		(SEE NOTES 1,2 BELOW).
С	= 4	IF THE DERIVATIVE OF THE
SOLUTION		
C		WITH RESPECT TO THETA IS
SPECIFIED		
C		AT THETA = TS (SEE NOTE
1 BELOW) AND		
C		SOLUTION IS SPECIFIED AT
THETA = TF.		
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = TS = 0 AND THE
SOLUTION IS		
C		SPECIFIED AT THETA =
TF.		
C C	- 6	IF THE SOLUTION IS
UNSPECIFIED AT	- 0	IF THE SOLUTION IS
		THETA = TS = 0 AND THE
C		INEIA - IS - U AND INE
DERIVATIVE		
C DECEMBER TO THE T		OF THE SOLUTION WITH
RESPECT TO THETA		TO OPERATOR AS SUPER
C		IS SPECIFIED AT THETA =
TF		(677 11077 0 777 011)
C	_	(SEE NOTE 2 BELOW).
C	= /	IF THE SOLUTION IS
SPECIFIED AT		
C		THETA = TS AND THE
SOLUTION IS		
C		UNSPECIFIED AT THETA =
TF = PI.		
C	= 8	IF THE DERIVATIVE OF THE
SOLUTION		
C		WITH RESPECT TO THETA IS
SPECIFIED		
C		AT THETA = TS (SEE NOTE
1 BELOW)		
C		AND THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = TF = PI.
C	= 9	IF THE SOLUTION IS
UNSPECIFIED AT		

```
THETA = TS = 0 AND THETA
= TF = PI.
С
С
                           NOTE 1:
                           IF TS = 0, DO NOT USE MBDCND
С
= 3,4, OR 8,
                           BUT INSTEAD USE MBDCND = 5,6,
OR 9 .
С
С
                           NOTE 2:
                           IF TF = PI, DO NOT USE MBDCND
= 2,3, OR 6,
                           BUT INSTEAD USE MBDCND = 7.8,
OR 9 .
С
С
                         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 =
C
3,4, OR 8,
C
                            BDTS(J) =
(D/DTHETA)U(TS,R(J)),
                             J = 1, 2, ..., N+1.
С
С
С
                          WHEN MBDCND HAS ANY OTHER
VALUE, BDTS IS
                          A DUMMY VARIABLE.
C
С
С
                         BDTF
C
                           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,
C
                           BDTF(J) =
```

```
(D/DTHETA)U(TF,R(J)),
                           J = 1, 2, ..., N+1.
С
C
C
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDTF IS
                           A DUMMY VARIABLE.
С
С
                         RS, RF
C
                            THE RANGE OF R, I.E., RS .LE.
R .LT. RF.
                           RS MUST BE LESS THAN RF. RS
MUST BE
C
                           NON-NEGATIVE.
С
С
                         Ν
С
                            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) DR+RS
                           FOR J = 1, 2, ..., N+1, WHERE DR
= (RF-RS)/N
С
                            IS THE PANEL WIDTH.
С
                            N MUST BE GREATER THAN 2
С
С
                         NBDCND
                            INDICATES THE TYPE OF
BOUNDARY CONDITION
                           AT R = RS AND R = RF.
C
С
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
                                 R = RS AND R = RF.
С
                           = 2 IF THE SOLUTION IS
SPECIFIED AT
                                 R = RS AND THE
DERIVATIVE
                                 OF THE SOLUTION WITH
RESPECT TO R
С
                                 IS SPECIFIED AT R = RF.
                           = 3 IF THE DERIVATIVE OF THE
```

SOLUTION		
С		WITH RESPECT TO R IS
SPECIFIED AT		
С		R = RS AND R = RF.
C	= 4	IF THE DERIVATIVE OF THE
SOLUTION		
C		WITH RESPECT TO R IS
SPECIFIED AT		
C		RS AND THE SOLUTION IS
SPECIFIED AT		
C		R = RF.
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		
C		R = RS = 0 (SEE NOTE
BELOW) AND THE		
C		SOLUTION IS SPECIFIED AT
R = RF.	_	TE MUE COLUMNON TO
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT		D - DC - 0 /CEE NOWE
C DELOWY AND THE		R = RS = 0 (SEE NOTE
BELOW) AND THE		
SOLUTION WITH		DERIVATIVE OF THE
C SOLUTION WITH		RESPECT TO R IS
SPECIFIED AT R = RF.		RESPECT TO R 15
C C		
C	NOTE	•
C		ND = 5 OR 6 CANNOT BE
USED WITH	NDDC	
C	MBDC	PND = 1, 2, 4, 5, OR 7. THE
FORMER	1220	
C	INDI	CATES THAT THE SOLUTION
IS UNSPECIFIED		
C	AT R	x = 0, THE LATTER
INDICATES THAT THE		•
C	SOLU	TION IS SPECIFIED).
C	USE INSTEAD NBDCND = 1 OR 2	
С		
С	BDRS	
С	A ON	E-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT		
C	SPEC	IFIES THE VALUES OF THE
DERIVATIVE OF		

```
THE SOLUTION WITH RESPECT TO
R AT R = RS.
C
С
                           WHEN NBDCND = 3 OR 4,
С
                             BDRS(I) =
(D/DR)U(THETA(I),RS),
                             I = 1, 2, ..., M+1.
С
С
                           WHEN NBDCND HAS ANY OTHER
VALUE, BDRS IS
                           A DUMMY VARIABLE.
C
С
                         BDRF
С
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1
                           THAT SPECIFIES THE VALUES OF
THE
                           DERIVATIVE OF THE SOLUTION
C
WITH RESPECT
                           TO R AT R = RF.
С
С
                           WHEN NBDCND = 2,3, OR 6,
                             BDRF(I) =
(D/DR)U(THETA(I),RF),
                             I = 1, 2, ..., M+1.
С
                           WHEN NBDCND HAS ANY OTHER
VALUE, BDRF IS
                           A DUMMY VARIABLE.
С
С
                         ELMBDA
C
                           THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                           EQUATION. IF LAMBDA .GT. 0,
A SOLUTION
С
                           MAY NOT EXIST. HOWEVER,
HWSCSP WILL
                           ATTEMPT TO FIND A SOLUTION.
IF NBDCND = 5
                           OR 6 OR MBDCND = 5, 6, 7, 8, OR
C
9, ELMBDA
                           MUST BE ZERO.
C
С
```

C	A TWO-DI	MENSIONAL ARRAY, OF
DIMENSION AT C	LEAST (M	I+1) * (N+1), SPECIFYING
VALUES OF THE		
C EQUATION AND	RIGHT SI	DE OF THE HELMHOLTZ
C	BOUNDARY	VALUES (IF ANY).
C	ON MILE T	NUMEDIOD E TO DEETNED
AS FOLLOWS:	ON THE I	NTERIOR, F IS DEFINED
C	FOR I =	2,3,,M AND J =
2,3,,N C	F(T.J) =	F (THETA(I),R(J)).
C	1 (1/0)	1 (1111111(1) / 11(0) / .
C DEFINED AS FOLLOWS:	ON THE B	BOUNDARIES, F IS
C	FOR J=1,	2,,N+1,
I=1,2,,M+1,		
C	MBDCND	F(1,J)
F(M+1,J)		- (-, -,
C		
C		
C	1	U(TS,R(J))
U(TF,R(J)) C	2	U(TS,R(J))
F(TF,R(J))		
C F(TF, R(J))	3	F(TS,R(J))
C	4	F(TS,R(J))
U(TF,R(J)) C	5	F(0,R(J))
U(TF,R(J))	5	F (U, K(U))
C	6	F(0,R(J))
F(TF,R(J)) C	7	U(TS,R(J))
F(PI,R(J))		
C F(PI,R(J))	8	F(TS,R(J))
C	9	F(0,R(J))
F(PI,R(J))		
C	NBDCND	F(I,1)

F(I,N+1)		
C		
С		
С	1	U(THETA(I),RS)
U(THETA(I),RF)		
С	2	U(THETA(I),RS)
F(THETA(I),RF)		
С	3	F(THETA(I),RS)
F(THETA(I),RF)		
C	4	F(THETA(I),RS)
U(THETA(I),RF)		
C	5	F(TS,0)
U(THETA(I),RF)		
C	6	F(TS,0)
F(THETA(I),RF)		, ,
C		
C	NOTE:	
C		BLE CALLS FOR BOTH
THE SOLUTION		
C	U AND THE	RIGHT SIDE F AT A
CORNER THEN		
C	THE SOLUT	ION MUST BE
SPECIFIED.	1111 001011	101, 11001 25
C		
C	IDIMF	
C		OR FIRST) DIMENSION
OF THE ARRAY	11111 1(0)// ((on ringry birmingron
C C	ב אכ דיי או	PPEARS IN THE
PROGRAM CALLING	I AD II AI	TIDAKS IN IIID
C CALLLING	писсер г	THIS PARAMETER IS
USED TO SPECIFY	INSCSP.	INIS FARAMETER IS
	חווה נוארד או	DIE DIMENSION OF F
C TRIME MIGH	THE VARIA	BLE DIMENSION OF F.
IDIMF MUST		S. D. A. L. 1
C	BE AT LEAS	ST M+1 .
С		
С	W	00 1 1 1
C	A fortran	90 derived TYPE
(fishworkspace) variable		
С	that must	be declared by the
user. The first		
С	declarativ	ve statement in the
user program		
С	calling HV	WSCSP must be:

C	
C	USE fish
C	
С	The declarative statement
C	
C	TYPE (fishworkspace) ::
W	
C	
C	must also be include in the
user program.	
C	The first statement makes the
fishpack module	
C	defined in the file "fish.f"
available to the	
C	user program calling HWSCSP.
The second statement	
C	declares a derived type
variable (defined in	
C	the module "fish.f") which is
used internally	
C	in HWSCSP to dynamically
allocate real and complex	
C	work space used in solution.
An error flag	
C	(IERROR = 20) is set if the
required work space	
С	allocation fails (for example
if N,M are too large)	
C	Real and complex values are
set in the components	
C	of W on a initial (INTL=0)
call to HWSCSP. These	
C	must be preserved on non-
initial calls (INTL=1)	
C	to HWSCSP. This eliminates
redundant calculations	
C	and saves compute time.
C ****	IMPORTANT! The user program
calling HWSCSP should	
С	include the statement:
С	
С	CALL FISHFIN(W)
C	
C	after the final approximation

is generated by C	HWSCSP. The will deallocate
the real and complex	imedal. Ind will dodiedade
C	work space of W. Failure to
include this statement	-
С	could result in serious
memory leakage.	
C	
C C CN CHEDITE	П
C ON OUTPUT	F COMMATMS THE COLUMNOM II/T TO
OF THE FINITE	CONTAINS THE SOLUTION U(I,J)
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	DITTELLATION THE INCOME THE PROPERTY OF THE PR
C	(THETA(I),R(J)), I =
1,2,,M+1,	
С	J =
1,2,,N+1 .	
С	
C	PERTRB
C OR DERIVATIVE	IF A COMBINATION OF PERIODIC
C C	BOUNDARY CONDITIONS IS
SPECIFIED FOR A	DOGNDIAN CONDITIONS 15
C	POISSON EQUATION (LAMBDA =
0), A SOLUTION	-
С	MAY NOT EXIST. PERTRB IS A
CONSTANT,	
C	CALCULATED AND SUBTRACTED
FROM F, WHICH	
C EXISTS. HWSCSP	ENSURES THAT A SOLUTION
C rwscsr	THEN COMPUTES THIS SOLUTION,
WHICH IS A	THEN COMPUTED THIS SCHOTTON,
C	LEAST SQUARES SOLUTION TO THE
ORIGINAL	~
С	APPROXIMATION. THIS SOLUTION
IS NOT UNIQUE	
С	AND IS UNNORMALIZED. THE
VALUE OF PERTRB	
C THE DICHE SIDE	SHOULD BE SMALL COMPARED TO
THE RIGHT SIDE	F. OTHERWISE , A SOLUTION IS
OBTAINED TO	r. OHIERWISE , A SOLUTION IS

С	AN ESSENTIALLY DIFFERENT		
PROBLEM. THIS			
C	COMPARISON SHOULD ALWAYS BE		
MADE TO INSURE			
C	THAT A MEANINGFUL SOLUTION		
HAS BEEN OBTAINED.			
C			
С	IERROR		
C	AN ERROR FLAG THAT INDICATES		
INVALID INPUT			
С	PARAMETERS. EXCEPT FOR		
NUMBERS 0 AND 10,			
C	A SOLUTION IS NOT ATTEMPTED.		
C			
C	= 1 TS.LT.O. OR TF.GT.PI		
C	= 2 TS.GE.TF		
C	= 3 M.LT.5		
C	= 4 MBDCND.LT.1 OR		
MBDCND.GT.9	1 122012.21.1		
C	= 5 RS.LT.0		
C	= 6 RS.GE.RF		
C	= 7 N.LT.5		
C	= 8 NBDCND.LT.1 OR		
NBDCND.GT.6	- 0 NDDCND.HI.I OK		
C	= 9 ELMBDA.GT.0		
C	= 10 IDIMF.LT.M+1		
C	= 11 ELMBDA.NE.O AND		
MBDCND.GE.5	- II ELMDDA.NE.O AND		
C C	= 12 ELMBDA.NE.O AND NBDCND		
EQUALS 5 OR 6	- IZ ELMDDA.NE.O AND NDDCND		
C C	= 13 MBDCND EQUALS 5,6 OR 9		
	- 13 MBDCND EQUALS 3,0 OK 9		
AND TS.NE.0	- 14 MODOND CE 7 AND ME NE DI		
C	= 14 MBDCND.GE.7 AND TF.NE.PI		
C FOLIAT C 3 4 OD 0	= 15 TS.EQ.O AND MBDCND		
EQUALS 3,4 OR 8	16 ME EO DI AND MODONO		
C	= 16 TF.EQ.PI AND MBDCND		
EQUALS 2,3 OR 6	17 170000 00 5 110 00 110 0		
C	= 17 NBDCND.GE.5 AND RS.NE.0		
C	= 18 NBDCND.GE.5 AND MBDCND		
EQUALS 1,2,4,5 OR	00 75 11		
C	= 20 If the dynamic		
allocation of real and			
C	complex work space in		
the derived type			

```
(fishworkspace) variable
W fails (e.g.,
                                if N,M are too large for
the platform used)
                          SINCE THIS IS THE ONLY MEANS
С
OF INDICATING
                          A POSSLIBY INCORRECT CALL TO
HWSCSP, THE
                          USER SHOULD TEST IERROR AFTER
A CALL.
C
С
                           The derived type
(fishworkspace) variable W
                          contains real and complex
values that must not
                          be destroyed if HWSCSP is
called again with
                           INTL=1.
C SPECIAL CONDITIONS
                        NONE
C I/O
                        NONE
C PRECISION
                       SINGLE
C REQUIRED files fish.f,blktri.f,comf.f
C LANGUAGE
                        FORTRAN 90
C HISTORY
                        WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE
                         1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE
                        LIBRARIES IN JANUARY 1980.
Revised by John
                        Adams in June 2004 using
Fortran 90 dynamically
                        allocated work space and
derived datat types
                        to eliminate mixed mode
conflicts in the earlier
                        versions.
```

C PORTABILITY FORTRAN 90 C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE EQUATIONS, INCORPORATES BOUNDARY DATA, AND ADJUSTS THE RIGHT SIDE OF SINGULAR SYSTEMS AND THEN CALLS BLKTRI TO SOLVE THE SYSTEM. C REFERENCES SWARZTRAUBER, P. AND R. SWEET, "EFFICIENT FORTRAN SUBPROGRAMS FOR THE SOLUTION OF ELLIPTIC EQUATIONS" С NCAR TN/IA-109, JULY, 1975, 138 PP. ******

HWSCYL

```
all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
```

```
С
C
                      the National Science Foundation
С
С
      SUBROUTINE HWSCYL
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                          ELMBDA, F, IDIMF, PERTRB, IERROR)
С
С
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N+1)
C ARGUMENTS
С
C LATEST REVISION June 2004
C PURPOSE
                          SOLVES A FINITE DIFFERENCE
APPROXIMATION
                          TO THE HELMHOLTZ EQUATION IN
CYLINDRICAL
                          COORDINATES. THIS MODIFIED
HELMHOLTZ EQUATION
С
                             (1/R) (D/DR) (R(DU/DR)) +
(D/DZ)(DU/DZ)
С
С
                             + (LAMBDA/R**2)U = F(R,Z)
С
С
                          RESULTS FROM THE FOURIER
TRANSFORM OF THE
                           THREE-DIMENSIONAL POISSON
EQUATION.
С
C USAGE
                           CALL HWSCYL
(A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
С
                                         PERTRB, IERROR, W)
С
C ARGUMENTS
```

C ON INPUT	A, B
C	THE RANGE OF R, I.E., A .LE.
R .LE. B.	·
С	A MUST BE LESS THAN B AND A
MUST BE	
С	NON-NEGATIVE.
C	
C	M
C WHICH THE	THE NUMBER OF PANELS INTO
C C	INTERVAL (A,B) IS SUBDIVIDED.
HENCE,	INTERVALI (A, D) IS SODDIVIDED.
C	THERE WILL BE M+1 GRID POINTS
IN THE	
С	R-DIRECTION GIVEN BY R(I) =
A+(I-1)DR,	
С	FOR $I = 1, 2,, M+1$, WHERE DR
= (B-A)/M	
C	IS THE PANEL WIDTH. M MUST BE
GREATER	muzzu O
C	THAN 3.
C	MRDCND
C	MBDCND TNDICATES THE TYPE OF
С	MBDCND INDICATES THE TYPE OF
C BOUNDARY CONDITIONS	INDICATES THE TYPE OF
C BOUNDARY CONDITIONS C	INDICATES THE TYPE OF
C BOUNDARY CONDITIONS C C C SPECIFIED AT	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS
C BOUNDARY CONDITIONS C C C SPECIFIED AT C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B.
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B.
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C C SPECIFIED AT C OF THE	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C OF THE C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C C SPECIFIED AT C OF THE	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C C SPECIFIED AT C R IS	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO
C BOUNDARY CONDITIONS C C C C SPECIFIED AT C C SPECIFIED AT C C SPECIFIED AT C OF THE C R IS C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO SPECIFIED AT R = B.
C BOUNDARY CONDITIONS C C C C SPECIFIED AT C C S C S C S C S C S C S C S C S C S	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO SPECIFIED AT R = B.
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C C SPECIFIED AT C OF THE C R IS C C SOLUTION	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO SPECIFIED AT R = B. = 3 IF THE DERIVATIVE OF THE WITH RESPECT TO R IS
C BOUNDARY CONDITIONS C C C C SPECIFIED AT C C SPECIFIED AT C OF THE C R IS C C SOLUTION C SPECIFIED AT C	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO SPECIFIED AT R = B. = 3 IF THE DERIVATIVE OF THE
C BOUNDARY CONDITIONS C C C SPECIFIED AT C C SPECIFIED AT C OF THE C R IS C C SOLUTION C SPECIFIED AT	INDICATES THE TYPE OF AT R = A AND R = B. = 1 IF THE SOLUTION IS R = A AND R = B. = 2 IF THE SOLUTION IS R = A AND THE DERIVATIVE SOLUTION WITH RESPECT TO SPECIFIED AT R = B. = 3 IF THE DERIVATIVE OF THE WITH RESPECT TO R IS

SOLUTION	
C	WITH RESPECT TO R IS
SPECIFIED AT	
С	R = A (SEE NOTE BELOW)
AND THE	
С	SOLUTION IS SPECIFIED AT
R = B.	
C	= 5 IF THE SOLUTION IS
UNSPECIFIED AT	
C	R = A = 0 AND THE
	IN - A - O AND THE
SOLUTION IS	
C	SPECIFIED AT R = B.
C	= 6 IF THE SOLUTION IS
UNSPECIFIED AT	
C	R = A = 0 AND THE
DERIVATIVE OF THE	
C	SOLUTION WITH RESPECT TO
R IS SPECIFIED	
С	AT $R = B$.
С	
C	IF $A = 0$, DO NOT USE MBDCND =
3 OR 4,	·
C	BUT INSTEAD USE MBDCND =
1,2,5, OR 6 .	
C	
C	BDA
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1 THAT	A ONE-DIMENSIONAL ARRAI OF
	CDECTETES MILE VALUES OF MILE
C	SPECIFIES THE VALUES OF THE
DERIVATIVE OF	
C	THE SOLUTION WITH RESPECT TO
R AT R = A.	
C	
C	WHEN MBDCND = $3 \text{ OR } 4$,
C	BDA(J) = (D/DR)U(A,Z(J)), J
= 1, 2,, N+1.	
С	
С	WHEN MBDCND HAS ANY OTHER
VALUE, BDA IS	
C	A DUMMY VARIABLE.
C	
C	BDB
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1 THAT	11 OND DIVIDIONAL WIGHT OF
THILL NAT TUNT	

```
SPECIFIES THE VALUES OF THE
DERIVATIVE
                            OF THE SOLUTION WITH RESPECT
TO R AT R = B.
С
                            WHEN MBDCND = 2,3, OR 6,
С
                             BDB(J) = (D/DR)U(B,Z(J)), J
= 1, 2, ..., N+1.
С
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDB IS
                           A DUMMY VARIABLE.
С
С
                         C,D
С
                           THE RANGE OF Z, I.E., C .LE.
Z .LE. D.
С
                           C MUST BE LESS THAN D.
С
С
                         Ν
С
                            THE NUMBER OF PANELS INTO
WHICH THE
                            INTERVAL (C, D) IS SUBDIVIDED.
HENCE,
                            THERE WILL BE N+1 GRID POINTS
IN THE
С
                           Z-DIRECTION GIVEN BY Z(J) =
C+(J-1)DZ,
                           FOR J = 1, 2, ..., N+1,
С
                            WHERE DZ = (D-C)/N IS THE
С
PANEL WIDTH.
C
                           N MUST BE GREATER THAN 3.
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
С
                           AT Z = C AND Z = D.
С
                           = 0 IF THE SOLUTION IS
C
PERIODIC IN Z,
                                 I.E., U(I,1) = U(I,N+1).
C
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
C
                                 Z = C AND Z = D.
                            = 2 IF THE SOLUTION IS
```

SPECIFIED AT	
C	Z = C AND THE DERIVATIVE
OF	
C	THE SOLUTION WITH
RESPECT TO Z IS	
С	SPECIFIED AT $Z = D$.
С	= 3 IF THE DERIVATIVE OF THE
SOLUTION	0
C	WITH RESPECT TO Z IS
C	SPECIFIED AT Z = C AND Z
	SPECIFIED AT Z - C AND Z
= D.	4
C	= 4 IF THE DERIVATIVE OF THE
SOLUTION	
C	WITH RESPECT TO Z IS
SPECIFIED AT	
C	Z = C AND THE SOLUTION
IS SPECIFIED	
C	AT $Z = D$.
С	
С	BDC
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT	
C	SPECIFIES THE VALUES OF THE
DERIVATIVE	SINCIPIES THE VALUES OF THE
	OF MIE COLUMNON MIMI DECDECM
C	OF THE SOLUTION WITH RESPECT
TO Z AT $Z = C$.	
С	
C	WHEN NBDCND = $3 \text{ OR } 4$,
C	BDC(I) = (D/DZ)U(R(I),C), I
$= 1, 2, \ldots, M+1.$	
C	
C	WHEN NBDCND HAS ANY OTHER
VALUE, BDC IS	
C	A DUMMY VARIABLE.
С	
C	BDD
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT	
C	SPECIFIES THE VALUES OF THE
DERIVATIVE OF	SIECTITES THE VALUES OF THE
	MILE COLLIMION MITHLE DECDEOM MO
C	THE SOLUTION WITH RESPECT TO
Z AT Z = D.	
C	
C	WHEN NBDCND = $2 \text{ OR } 3$,

```
BDD(I) = (D/DZ)U(R(I),D), I
= 1, 2, ..., M+1
С
C
                            WHEN NBDCND HAS ANY OTHER
VALUE, BDD IS
                            A DUMMY VARIABLE.
С
С
                          ELMBDA
C
                            THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                            EQUATION. IF LAMBDA .GT. 0,
A SOLUTION
                            MAY NOT EXIST. HOWEVER,
HWSCYL WILL
                            ATTEMPT TO FIND A SOLUTION.
LAMBDA MUST
                            BE ZERO WHEN MBDCND = 5 OR 6
С
С
                            A TWO-DIMENSIONAL ARRAY, OF
DIMENSION AT
                            LEAST (M+1) * (N+1), SPECIFYING
C
VALUES
                            OF THE RIGHT SIDE OF THE
HELMHOLTZ
                            EQUATION AND BOUNDARY DATA
(IF ANY).
С
                            ON THE INTERIOR, F IS DEFINED
AS FOLLOWS:
                            FOR I = 2, 3, \ldots, M AND J =
2,3,...,N
                            F(I,J) = F(R(I),Z(J)).
С
                            ON THE BOUNDARIES F IS
DEFINED AS FOLLOWS:
                            FOR J = 1, 2, ..., N+1 AND I =
1, 2, \ldots, M+1
C
                            MBDCND F(1,J)
F(M+1,J)
С
```

C	1	ΙΙ / 7
U(B, Z(J))	1	U(A,Z(J))
C	2	U(A,Z(J))
F(B,Z(J))		
C F(B, Z(J))	3	F(A,Z(J))
C	4	F(A,Z(J))
U(B,Z(J))		
C	5	F(0,Z(J))
U(B,Z(J)) C	6	F(0,Z(J))
F(B, Z(J))	Ŭ.	1 (0/2(0))
С		
C E (T. N. 1)	NBDCND	F(I,1)
F(I,N+1)		
С		_ (_ (_)
C F(R(I),C)	0	F(R(I),C)
C (K(1),C)	1	U(R(I),C)
U(R(I),D)		
C	2	U(R(I),C)
F(R(I),D) C	3	F(R(I),C)
F(R(I),D)	9	
C	4	F(R(I),C)
U(R(I),D)		
C	NOTE:	
C		TABLE CALLS FOR BOTH
THE SOLUTION		
C CORNER THEN	U AND TH	HE RIGHT SIDE F AT A
C	THE SOLU	JTION MUST BE
SPECIFIED.		
C		
C	IDIMF THE ROW	(OR FIRST) DIMENSION
OF THE ARRAY	11111 11011	(OIC LINGL) DIFINITION
С	F AS IT	APPEARS IN THE
PROGRAM CALLING	ITAOONT	
C	HWSCYL.	THIS PARAMETER IS

USED TO SPECIFY	
C	THE VARIABLE DIMENSION OF F.
IDIMF MUST	
C	BE AT LEAST M+1 .
C	
C	
C ON OUTPUT	F
C	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	
C DIE COLD DOINE	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	/D/T\ F/T\\ T 1 0 MI1
C	(R(I),Z(J)), I = 1,2,,M+1,
J = 1, 2,, N+1.	
C	PERTRB
C	IF ONE SPECIFIES A
COMBINATION OF PERIODIC,	IF ONE SPECIFIES A
C.	DERIVATIVE, AND UNSPECIFIED
BOUNDARY	DUNITURE, TAND CHOIDCILLED
C	CONDITIONS FOR A POISSON
EQUATION	CONDITIONS FOR 11 FOLDSON
C	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
C	PERTRB IS A CONSTANT,
CALCULATED AND	,
C	SUBTRACTED FROM F, WHICH
ENSURES THAT A	·
C	SOLUTION EXISTS. HWSCYL THEN
COMPUTES	
C	THIS SOLUTION, WHICH IS A
LEAST SQUARES	
C	SOLUTION TO THE ORIGINAL
APPROXIMATION.	
C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	
C	A SOLUTION. HENCE, THE
SOLUTION IS NOT	
C	UNIQUE. THE VALUE OF PERTRB
SHOULD BE	
C	SMALL COMPARED TO THE RIGHT
SIDE F.	OFFICE A COLUMN TO
C ODMATNED TO AN	OTHERWISE, A SOLUTION IS
OBTAINED TO AN	
C	ESSENTIALLY DIFFERENT

PROBLEM. THIS	
C	COMPARISON SHOULD ALWAYS BE
MADE TO INSURE	
C	THAT A MEANINGFUL SOLUTION
HAS BEEN OBTAINED.	
C	
	TEDDOD
C	IERROR
C	AN ERROR FLAG WHICH INDICATES
INVALID INPUT	
C	PARAMETERS. EXCEPT FOR
NUMBERS 0 AND 11,	
C	A SOLUTION IS NOT ATTEMPTED.
C	TO CONCINCT TO THE TIME TO .
	O NO EDDOD
C	= 0 NO ERROR.
С	= 1 A .LT. 0 .
С	= 2 A .GE. B.
C	= 3 MBDCND .LT. 1 OR MBDCND
.GT. 6 .	
C	= 4 C .GE. D.
C	= 5 N .LE. 3
C	
	= 6 NBDCND .LT. 0 OR NBDCND
.GT. 4 .	
C	= 7 A $=$ 0, MBDCND $=$ 3 OR 4
•	
C	= 8 A .GT. 0, MBDCND .GE. 5
	·
C	= 9 A $=$ 0, LAMBDA .NE. 0,
MBDCND .GE. 5 .	<i>5</i> 11 0, manbh 111. 0,
	- 10 IDIME IE MI1
C	= 10 IDIMF .LT. M+1 .
C	= 11 LAMBDA .GT. 0 .
C	= 12 M .LE. 3
C	= 20 If the dynamic
allocation of real and	
C	complex work space
required for solution	compion worm space
_	foils /for onemals if
C	fails (for example if
N,M are too large	
C	for your computer)
C	
C	SINCE THIS IS THE ONLY MEANS
OF INDICATING	
C	A POSSIBLY INCORRECT CALL TO
	W LOSSIBIL LINCOUNTEL CATH IO
HWSCYL, THE	HORD (HOLL) TO THE TOTAL TOTAL
C	USER SHOULD TEST IERROR AFTER

```
THE CALL.
С
C SPECIAL CONDITIONS NONE
C I/O
                      NONE
C PRECISION
               SINGLE
C REQUIRED files
fish.f,genbun.f,gnbnaux.f,comf.f
C LANGUAGE
                      FORTRAN 90
C HISTORY
                      WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE
                      1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE
                       LIBRARIES IN JANUARY 1980.
                       Revised in June 2004 by John
Adams using
                      Fortran 90 dynamically
allocated work space.
С
C PORTABILITY FORTRAN 90
C ALGORITHM
                      THE ROUTINE DEFINES THE FINITE
DIFFERENCE
                       EQUATIONS, INCORPORATES
BOUNDARY DATA, AND
                       ADJUSTS THE RIGHT SIDE OF
SINGULAR SYSTEMS
                       AND THEN CALLS GENBUN TO SOLVE
THE SYSTEM.
С
C TIMING
                      FOR LARGE M AND N, THE
OPERATION COUNT
                       IS ROUGHLY PROPORTIONAL TO
С
                         M*N* (LOG2 (N)
                       BUT ALSO DEPENDS ON INPUT
PARAMETERS NBDCND
С
                       AND MBDCND.
```

C ACCURACY THE SOLUTION PROCESS EMPLOYED RESULTS IN A LOSS OF NO MORE THAN THREE SIGNIFICANT DIGITS FOR N AND M AS LARGE AS 64. MORE DETAILS ABOUT ACCURACY CAN BE FOUND IN THE DOCUMENTATION FOR SUBROUTINE GENBUN WHICH IS THE ROUTINE THAT SOLVES THE FINITE DIFFERENCE EQUATIONS. C REFERENCES SWARZTRAUBER, P. AND R. SWEET, "EFFICIENT FORTRAN SUBPROGRAMS FOR THE SOLUTION OF ELLIPTIC EQUATIONS" CNCAR TN/IA-109, JULY, 1975, 138 PP. C**************** ******

HWSPLR

```
Research
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
С
С
                        A Package of Fortran 77 and 90
С
                       Subroutines and Example Programs
С
С
С
                      for Modeling Geophysical Processes
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
```

```
С
С
                      the National Science Foundation
С
С
      SUBROUTINE HWSPLR
С
(A, B, M, MBDCND, BDA, BDB, C, D, N, NBDCND, BDC, BDD,
                          ELMBDA, F, IDIMF, PERTRB, IERROR)
С
С
C DIMENSION OF
BDA(N), BDB(N), BDC(M), BDD(M), F(IDIMF, N+1)
C ARGUMENTS
C
C LATEST REVISION
                         June 2004
C PURPOSE
                          SOLVES A FINITE DIFFERENCE
APPROXIMATION TO
                          THE HELMHOLTZ EQUATION IN POLAR
COORDINATES.
С
                          THE EQUATION IS
С
С
                               (1/R)(D/DR)(R(DU/DR)) +
(1/R**2) (D/DTHETA) (DU/DTHETA) +
С
                               LAMBDA*U = F(R, THETA).
С
C USAGE
                          CALL HWSPLR
(A,B,M,MBDCND,BDA,BDB,C,D,N,
NBDCND, BDC, BDD, ELMBDA, F, IDIMF,
С
                                        PERTRB, IERROR, W)
С
C ARGUMENTS
C ON INPUT
                          A,B
                             THE RANGE OF R, I.E., A .LE.
R .LE. B.
C
                            A MUST BE LESS THAN B AND A
MUST BE
```

С	NON-NEG	GATIVE.
С		
C	M	
C	THE NUM	BER OF PANELS INTO
WHICH THE		
С	INTERVA	AL (A,B) IS SUBDIVIDED.
HENCE,		
C	THERE W	VILL BE M+1 GRID POINTS
IN THE		
C	R-DTRF.C	CTION GIVEN BY R(I) =
A+(I-1)DR,	1. 51100	
C C	FOR T =	= 1,2,,M+1,
C		DR = (B-A)/M IS THE
PANEL WIDTH.	AATTITUTE T), - (D V)\II IO IIII
C C	M MICH	BE GREATER THAN 3.
	M MOST	DE GREATER IDAN 3.
C	MDDCVID	
C	MBDCND	
C CONDITION	TNDTCAT	ES THE TYPE OF
BOUNDARY CONDITION	7 111 1	
C	A.I. K =	A AND R = B.
C	_	
C	= 1 IF	THE SOLUTION IS
SPECIFIED AT		
C		= A AND R $=$ B.
С	= 2 IF	THE SOLUTION IS
SPECIFIED AT		
С	R	= A AND THE DERIVATIVE
OF		
C	TH	E SOLUTION WITH
RESPECT TO R IS		
C	SF	PECIFIED AT $R = B$.
C	= 3 IF	THE DERIVATIVE OF THE
SOLUTION		
С	WI	TH RESPECT TO R IS
SPECIFIED AT		
C	R	= A (SEE NOTE BELOW)
AND $R = B$.	- 1	(======================================
C E.	= 4 TF	THE DERIVATIVE OF THE
SOLUTION	1 11	
C	TAT T	TH RESPECT TO R IS
SPECIFIED AT	VVI	
C C	D	- A (SEE MOTE DETOM)
	K	= A (SEE NOTE BELOW)
AND THE	0.0	NIIMION IC ODECTETED AM
C	SC	DLUTION IS SPECIFIED AT

```
R = B.
                           = 5 IF THE SOLUTION IS
С
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.
С
С
                           NOTE:
                           IF A = 0, DO NOT USE MBDCND =
3 OR 4, BUT
                           INSTEAD USE MBDCND = 1,2,5,
С
OR 6 .
C
С
                         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.
С
С
                           WHEN MBDCND = 3 OR 4,
                            BDA(J) =
(D/DR)U(A, THETA(J)),
                             J = 1, 2, ..., N+1.
С
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDA IS
С
                           A DUMMY VARIABLE.
С
С
                         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.
```

```
С
                           WHEN MBDCND = 2,3, OR 6,
                             BDB(J) =
(D/DR)U(B,THETA(J)),
                             J = 1, 2, ..., N+1.
С
C
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDB IS
С
                          A DUMMY VARIABLE.
С
С
                         C,D
C
                           THE RANGE OF THETA, I.E., C
.LE.
                           THETA .LE. D. C MUST BE LESS
THAN D.
С
С
                         N
С
                           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) DTHETA FOR
С
С
                           J = 1, 2, ..., N+1, WHERE
С
                           DTHETA = (D-C)/N IS THE PANEL
WIDTH.
                           N MUST BE GREATER THAN 3.
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITIONS
                           AT THETA = C AND AT THETA =
D.
С
                           = 0 IF THE SOLUTION IS
PERIODIC IN THETA,
                                 I.E., U(I,J) = U(I,N+J).
                           = 1 IF THE SOLUTION IS
SPECIFIED AT
С
                                THETA = C AND THETA = D
С
                                (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
С
                                (SEE NOTE BELOW).
                           = 4 IF THE DERIVATIVE OF THE
С
SOLUTION
                                WITH RESPECT TO THETA IS
SPECIFIED
                                AT THETA = C AND THE
SOLUTION IS
                                SPECIFIED AT THETA = D
С
                                (SEE NOTE BELOW).
С
С
                           NOTE:
                           WHEN NBDCND = 1,2, OR 4, DO
С
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
С
                         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
С
OR 4,
С
                            BDC(I) =
(D/DTHETA)U(R(I),C),
С
                             I = 1, 2, ..., M+1
С
                           WHEN NBDCND HAS ANY OTHER
```

VALUE, BDC IS C C	A DUMMY VARIABLE.
C	BDD A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT C	SPECIFIES THE VALUES OF THE
DERIVATIVE C	OF THE SOLUTION WITH RESPECT
TO THETA AT C OR 3,	THETA = D. WHEN NBDCND = 2
C C	BDD(I) =
(D/DTHETA) U(R(I),D), C C	I = 1, 2,, M+1.
C VALUE, BDD IS	WHEN NBDCND HAS ANY OTHER
C	A DUMMY VARIABLE.
C C	ELMBDA THE CONSTANT LAMBDA IN THE
HELMHOLTZ C A SOLUTION	EQUATION. IF LAMBDA .LT. 0,
C HWSPLR WILL	MAY NOT EXIST. HOWEVER,
C C	ATTEMPT TO FIND A SOLUTION.
C C	F A TWO-DIMENSIONAL ARRAY, OF
DIMENSION AT C VALUES	LEAST (M+1) * (N+1), SPECIFYING
C HELMHOLTZ	OF THE RIGHT SIDE OF THE
C (IF ANY).	EQUATION AND BOUNDARY DATA
C AS FOLLOWS:	ON THE INTERIOR, F IS DEFINED
C 2,3,,N	FOR $I = 2,3,,M$ AND $J =$

```
С
                            F(I,J) = F(R(I), THETA(J)).
С
C
                            ON THE BOUNDARIES F IS
DEFINED AS FOLLOWS:
                           FOR J = 1, 2, ..., N+1 AND I =
1, 2, \ldots, M+1
C
С
                           MBDCND F(1,J)
F(M+1,J)
C
С
                              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))
С
                              4
                                    F(A, THETA(J))
U(B, THETA(J))
                              5 F(0,0)
U(B, THETA(J))
                              6
                                    F(0,0)
F(B, THETA(J))
С
                           NBDCND
                                   F(I,1)
F(I,N+1)
С
С
С
                              0 	 F(R(I),C)
F(R(I),C)
                              1
                                    U(R(I),C)
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)
C
С
                           NOTE:
С
                            IF THE TABLE CALLS FOR BOTH
THE SOLUTION
```

C	U AND THE RIGHT SIDE F AT A
CORNER THEN	
C	THEN THE SOLUTION MUST BE
SPECIFIED.	
C	
C	IDIMF
C	THE ROW (OR FIRST) DIMENSION
OF THE ARRAY	
C	F AS IT APPEARS IN THE
PROGRAM CALLING	
С	HWSPLR. THIS PARAMETER IS
USED TO SPECIFY	
C	THE VARIABLE DIMENSION OF F.
IDIMF MUST	
C	BE AT LEAST M+1.
C	DE AI LEASI MII.
C ON OUTPUT	F
	_
C	CONTAINS THE SOLUTION U(I,J)
OF THE FINITE	
C	DIFFERENCE APPROXIMATION FOR
THE GRID POINT	
C	(R(I), THETA(J)),
C	I = 1, 2,, M+1, J =
1,2,,N+1 .	
C	
C	PERTRB
C	IF A COMBINATION OF PERIODIC,
DERIVATIVE,	
C	OR UNSPECIFIED BOUNDARY
CONDITIONS IS	
C	SPECIFIED FOR A POISSON
EQUATION	
C	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	·
C	PERTRB IS A CONSTANT,
CALCULATED AND	
C	SUBTRACTED FROM F, WHICH
ENSURES THAT A	SOBIRCIES TROPET, WILLOW
C C	SOLUTION EXISTS. HWSPLR THEN
	DOTICIN EXTRES . UMBETY IUFIN
COMPUTES	MILLO COLLIMION WILLOU TO A
C LEAGE GOLLADEG	THIS SOLUTION, WHICH IS A
LEAST SQUARES	GOLUMION TO THE OPERATOR
C APPROXIMATION.	SOLUTION TO THE ORIGINAL

C	THIS SOLUTION PLUS ANY
CONSTANT IS ALSO	
С	A SOLUTION. HENCE, THE
SOLUTION IS NOT	
С	UNIQUE. PERTRB SHOULD BE
SMALL COMPARED	
С	TO THE RIGHT SIDE. OTHERWISE,
A SOLUTION	
С	IS OBTAINED TO AN ESSENTIALLY
DIFFERENT	
С	PROBLEM. THIS COMPARISON
SHOULD ALWAYS	
С	BE MADE TO INSURE THAT A
MEANINGFUL	
С	SOLUTION HAS BEEN OBTAINED.
С	
С	IERROR
С	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
С	PARAMETERS. EXCEPT FOR
NUMBERS 0 AND 11,	
С	A SOLUTION IS NOT ATTEMPTED.
C	
C	= 0 NO ERROR.
C	= 1 A .LT. 0 .
C	= 2 A .GE. B.
C	= 3 MBDCND .LT. 1 OR MBDCND
.GT. 6 .	
C	= 4 C .GE. D.
C	= 5 N .LE. 3
С	= 6 NBDCND .LT. 0 OR .GT. 4
•	
С	= 7 A $=$ 0, MBDCND $=$ 3 OR 4
·	
С	= 8 A .GT. 0, MBDCND .GE. 5
·	- 0 MDDGND CE E NDDGND
C	= 9 MBDCND .GE. 5, NBDCND
.NE. 0 C	AND NIDDONID NIE 2
C	AND NBDCND .NE. 3 .
C	= 10 IDIMF .LT. M+1 .
C	= 11 LAMBDA .GT. 0 . = 12 M .LE. 3
C	
	= 20 If the dynamic
allocation of real and	

C	complex work space
required for solution	
C	fails (for example if
N,M are too large	
C	for your computer)
C	
С	SINCE THIS IS THE ONLY MEANS
OF INDICATING	
С	A POSSIBLY INCORRECT CALL TO
HWSPLR, THE	
C	USER SHOULD TEST IERROR AFTER
THE CALL.	
C	
	NONE
C C	IVOIVE
C I/O	NONE
C 1/0	NONE
C PRECISION	SINGLE
C	STINGTE
C REQUIRED files	££ £
fish.f,genbun.f,gnbnaux.	I, COMI.I
C	
C LANGUAGE	FORTRAN 90
C	
C HISTORY	WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE	
C	1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE	
C	LIBRARIES IN JANUARY 1980.
С	Revised in June 2004 by John
Adams using	
C	Fortran 90 dynamically
allocated work space.	
C	
C PORTABILITY	FORTRAN 90
С	
C ALGORITHM	THE ROUTINE DEFINES THE FINITE
DIFFERENCE	
С	EQUATIONS, INCORPORATES
BOUNDARY DATA, AND	
C	ADJUSTS THE RIGHT SIDE OF
SINGULAR SYSTEMS	-
C	AND THEN CALLS GENBUN TO SOLVE
THE SYSTEM.	

C TIMING FOR LARGE M AND N, THE OPERATION COUNT IS ROUGHLY PROPORTIONAL TO C С M*N* (LOG2 (N) C BUT ALSO DEPENDS ON INPUT PARAMETERS NBDCND AND MBDCND. С C ACCURACY THE SOLUTION PROCESS EMPLOYED RESULTS IN A LOSS OF NO MORE THAN THREE SIGNIFICANT DIGITS FOR N AND M AS LARGE AS 64. MORE DETAILS ABOUT ACCURACY CAN BE FOUND IN THE DOCUMENTATION FOR SUBROUTINE GENBUN WHICH IS THE ROUTINE THAT SOLVES THE FINITE DIFFERENCE EQUATIONS. C REFERENCES SWARZTRAUBER, P. AND R. SWEET, "EFFICIENT FORTRAN SUBPROGRAMS FOR THE SOLUTION OF ELLIPTIC EQUATIONS" С С NCAR TN/IA-109, JULY, 1975, 138 PP. *****

HWSSSP

C file hwsssp.txt (documentation for the FISHPACK solver HWSSSP)
C

```
С
                         copyright (c) 2005 by UCAR
C
С
              University Corporation for Atmospheric
Research
                              all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
С
```

```
C *
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
C
C
                          which is sponsored by
С
С
                     the National Science Foundation
С
С
      SUBROUTINE HWSSSP
(TS, TF, M, MBDCND, BDTS, BDTF, PS, PF, N, NBDCND, BDPS,
BDPF, ELMBDA, F, IDIMF, PERTRB, IERROR)
C DIMENSION OF
                         BDTS (N+1), BDTF (N+1),
BDPS (M+1), BDPF (M+1),
C ARGUMENTS
                         F(IDIMF, N+1)
C LATEST REVISION
                       June 2004
С
C PURPOSE
                         SOLVES A FINITE DIFFERENCE
APPROXIMATION TO
                         THE HELMHOLTZ EQUATION IN
SPHERICAL
                         COORDINATES AND ON THE SURFACE
OF THE UNIT
                         SPHERE (RADIUS OF 1). THE
EQUATION IS
С
(1/SIN(THETA)) (D/DTHETA) (SIN(THETA)
                            (DU/DTHETA)) +
(1/SIN(THETA)**2)(D/DPHI)
                            (DU/DPHI) + LAMBDA*U =
```

```
F (THETA, PHI)
С
C
                          WHERE THETA IS COLATITUDE AND
PHI IS
                          LONGITUDE.
C
С
C USAGE
                          CALL HWSSSP
(TS, TF, M, MBDCND, BDTS, BDTF, PS, PF,
N, NBDCND, BDPS, BDPF, ELMBDA, F,
IDIMF, PERTRB, IERROR, W)
C ARGUMENTS
C ON INPUT
                         TS, TF
С
                            THE RANGE OF THETA
(COLATITUDE), I.E.,
                           TS .LE. THETA .LE. TF. TS
MUST BE LESS
                            THAN TF. TS AND TF ARE IN
RADIANS.
                            A TS OF ZERO CORRESPONDS TO
C
THE NORTH
                            POLE AND A TF OF PI
CORRESPONDS TO
                            THE SOUTH POLE.
С
С
                            * * * IMPORTANT * * *
С
C.
                            IF TF IS EQUAL TO PI THEN IT
MUST BE
                            COMPUTED USING THE STATEMENT
C
                            TF = PIMACH(DUM). THIS
INSURES THAT TF
                            IN THE USER'S PROGRAM IS
EQUAL TO PI IN
                            THIS PROGRAM WHICH PERMITS
SEVERAL TESTS
                            OF THE INPUT PARAMETERS THAT
OTHERWISE
С
                            WOULD NOT BE POSSIBLE.
С
```

C	M
C	THE NUMBER OF PANELS INTO
WHICH THE	
C	INTERVAL (TS,TF) IS
SUBDIVIDED.	
C	HENCE, THERE WILL BE M+1 GRID
POINTS IN THE	
C	THETA-DIRECTION GIVEN BY
C	THETA(I) = $(I-1)$ DTHETA+TS FOR
C	$I = 1, 2, \dots, M+1, \text{ WHERE}$
C	
	DTHETA = (TF-TS)/M IS THE
PANEL WIDTH.	
C	M MUST BE GREATER THAN 5
C	
C	MBDCND
C	INDICATES THE TYPE OF
BOUNDARY CONDITION	
С	AT THETA = TS AND THETA = TF.
C	
C	= 1 IF THE SOLUTION IS
SPECIFIED AT	- 1 IF THE SOLUTION IS
С	THETA = TS AND THETA =
TF.	
C	= 2 IF THE SOLUTION IS
SPECIFIED AT	
С	THETA = TS AND THE
DERIVATIVE OF	
C	THE SOLUTION WITH
RESPECT TO THETA IS	
C	SPECIFIED AT THETA = TF
C	(SEE NOTE 2 BELOW).
C	= 3 IF THE DERIVATIVE OF THE
	- 3 IF INE DERIVATIVE OF THE
SOLUTION	
С	WITH RESPECT TO THETA IS
SPECIFIED	
C	SPECIFIED AT THETA = TS
AND	
C	THETA = TF (SEE NOTES
1,2 BELOW).	
C	= 4 IF THE DERIVATIVE OF THE
SOLUTION	
C	WITH RESPECT TO THETA IS
	MIIII VESTECT TO THEIR IS
SPECIFIED	AM MILLERA MC /COR NICES
C	AT THETA = TS (SEE NOTE

1 BELOW)		
C		AND THE SOLUTION IS
SPECIFIED AT		
C		THETA = TF.
C	= 5	IF THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = $TS = 0$ AND THE
SOLUTION		
C		IS SPECIFIED AT THETA =
TF.		
C	= 6	IF THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = $TS = 0$ AND THE
DERIVATIVE		
C		OF THE SOLUTION WITH
RESPECT TO THETA		
C		IS SPECIFIED AT THETA =
TF		
C		(SEE NOTE 2 BELOW).
C	= 7	IF THE SOLUTION IS
SPECIFIED AT	,	
C		THETA = TS AND THE
SOLUTION IS		THEIA - 15 AND THE
C C		IS UNSPECIFIED AT THETA
= TF = PI.		15 ONSFECTFIED AT THETA
- 1r - r1.	_ 0	IF THE DERIVATIVE OF THE
SOLUTION	- 0	IF INC DERIVATIVE OF THE
C		WITH RESPECT TO THETA IS
		WITH RESPECT TO THETA IS
SPECIFIED		AM MILEMA — MC /CEE NOME
C 1 DELOGA AND		AT THETA = TS (SEE NOTE
1 BELOW) AND		MILE COLUMNON TO
C		THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = TF = PI.
C	= 9	IF THE SOLUTION IS
UNSPECIFIED AT		
C		THETA = $TS = 0$ AND THETA
= TF = PI.		
C		
С	NOTE	S:
C	IF T	S = 0, DO NOT USE MBDCND
= 3,4, OR 8,		
C	BUT	INSTEAD USE MBDCND = $5,6,$
OR 9 .		

```
IF TF = PI, DO NOT USE MBDCND
С
= 2,3, OR 6,
                           BUT INSTEAD USE MBDCND = 7.8,
C
OR 9 .
C
С
                         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,
C
                           BDTS(J) =
(D/DTHETA)U(TS,PHI(J)),
                            J = 1, 2, ..., N+1.
С
С
                            WHEN MBDCND HAS ANY OTHER
VALUE, BDTS IS
                           A DUMMY VARIABLE.
C
С
С
                         BDTF
C
                            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,
C
                            BDTF(J) =
(D/DTHETA) U (TF, PHI (J)),
                            J = 1, 2, ..., N+1.
С
С
С
                            WHEN MBDCND HAS ANY OTHER
VALUE, BDTF IS
                            A DUMMY VARIABLE.
С
\mathsf{C}
                         PS, PF
                            THE RANGE OF PHI (LONGITUDE),
```

```
I.E.,
                           PS .LE. PHI .LE. PF. PS MUST
С
BE LESS
                           THAN PF. PS AND PF ARE IN
RADIANS.
                           IF PS = 0 AND PF = 2*PI,
PERIODIC
                           BOUNDARY CONDITIONS ARE
USUALLY PRESCRIBED.
С
                           * * * IMPORTANT * * *
С
C
                           IF PF IS EQUAL TO 2*PI THEN
IT MUST BE
                           COMPUTED USING THE STATEMENT
C
                           PF = 2.*PIMACH(DUM). THIS
INSURES THAT
                           PF IN THE USERS PROGRAM IS
EQUAL TO
                           2*PI IN THIS PROGRAM WHICH
PERMITS TESTS
                           OF THE INPUT PARAMETERS THAT
OTHERWISE
                           WOULD NOT BE POSSIBLE.
С
С
                         Ν
                           THE NUMBER OF PANELS INTO
WHICH THE
                           INTERVAL (PS, PF) IS
SUBDIVIDED.
                           HENCE, THERE WILL BE N+1 GRID
POINTS
С
                           IN THE PHI-DIRECTION GIVEN BY
                           PHI(J) = (J-1)DPHI+PS FOR
С
С
                           J = 1, 2, ..., N+1, WHERE
                           DPHI = (PF-PS)/N IS THE PANEL
С
WIDTH.
                           N MUST BE GREATER THAN 4
C
С
С
                         NBDCND
                           INDICATES THE TYPE OF
BOUNDARY CONDITION
С
                           AT PHI = PS AND PHI = PF.
С
```

C DEDUCATE THE DUT	= 0	IF THE SOLUTION IS
PERIODIC IN PHI,		I.U., U(I,J) = U(I,N+J).
C	= 1	IF THE SOLUTION IS
SPECIFIED AT	_	II III BOLOTION IB
C		PHI = PS AND PHI = PF
С		(SEE NOTE BELOW).
С	= 2	IF THE SOLUTION IS
SPECIFIED AT		
С		PHI = PS (SEE NOTE
BELOW)		
C		AND THE DERIVATIVE OF
THE SOLUTION C		WITH RESPECT TO PHI IS
SPECIFIED		WIIN RESPECT TO FREE TS
C		AT PHI = PF.
C	= 3	IF THE DERIVATIVE OF THE
SOLUTION		
С		WITH RESPECT TO PHI IS
SPECIFIED		
C		AT PHI = PS AND PHI =
PF.	4	
C	= 4	IF THE DERIVATIVE OF THE
SOLUTION C		WITH RESPECT TO PHI IS
SPECIFIED		WIIII NESIECI TO TIII IS
C		AT PS AND THE SOLUTION
IS SPECIFIED		
С		AT PHI = PF
С		
C	NOTE	
C	NBDC	ND = 1, 2, OR 4 CANNOT BE
USED WITH	MDDC	
C FORMER INDICATES	MBDC	ND = 5, 6, 7, 8, OR 9. THE
C C	ТНАТ	THE SOLUTION IS
SPECIFIED AT A POLE, THE	111111	
C	LATT	ER INDICATES THAT THE
SOLUTION IS NOT		
С	SPEC	IFIED. USE INSTEAD
MBDCND = 1 OR 2.		
C		
	BDPS	
C	A ON	E-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,
C
                            BDPS(I) =
(D/DPHI)U(THETA(I),PS),
                            I = 1, 2, ..., M+1
С
                          WHEN NBDCND HAS ANY OTHER
VALUE, BDPS IS
                          A DUMMY VARIABLE.
С
С
                        BDPF
С
                          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,
C
                            BDPF(I) =
(D/DPHI)U(THETA(I), PF),
                            I = 1, 2, ..., M+1
С
C
                          WHEN NBDCND HAS ANY OTHER
VALUE, BDPF IS
                          A DUMMY VARIABLE.
C
С
С
                        ELMBDA
C
                           THE CONSTANT LAMBDA IN THE
HELMHOLTZ
                          EQUATION. IF LAMBDA .GT. 0,
A SOLUTION
                          MAY NOT EXIST. HOWEVER,
HWSSSP WILL
С
                          ATTEMPT TO FIND A SOLUTION.
С
```

C	A TWO-DI	MENSIONAL ARRAY THAT
SPECIFIES THE	VATIIF ○F	THE RIGHT SIDE OF
THE HELMHOLTZ	VALOE OF	THE RIGHT SIDE OF
С	EQUATION	AND BOUNDARY VALUES
(IF ANY).	E MIICT E	BE DIMENSIONED AT
LEAST (M+1)*(N+1).	r MOSI L	DE DIFINITIONED AT
C		
C AC FOLLOWS:	ON THE I	INTERIOR, F IS DEFINED
AS FOLLOWS:	FOR I =	2,3,,M AND J =
2,3,,N		, , ,
C	F(I,J) =	= F(THETA(I), PHI(J)).
C	ON THE E	BOUNDARIES F IS
DEFINED AS FOLLOWS:		
C	FOR J =	1, 2,, N+1 AND I =
1,2,,M+1 C		
C	MBDCND	F(1,J)
F(M+1,J)		
C		
С		
C	1	U(TS, PHI(J))
U(TF, PHI(J)) C	2	U(TS,PHI(J))
F(TF,PHI(J))		- (- /)
C E (ME DIT (T))	3	F(TS,PHI(J))
F(TF, PHI(J))	4	F(TS,PHI(J))
U(TF,PHI(J))		, , , , ,
C	5	F(0,PS)
U(TF, PHI(J))	6	F(0,PS)
F(TF,PHI(J))		(-, -,
C E(DT DG)	7	U(TS, PHI(J))
F(PI, PS)	8	F(TS,PHI(J))
F(PI,PS)		
C C	9	F(0,PS)
F(PI, PS)		

C	NBDCND	F(I,1)
F(I,N+1)		
C		
C		
C	0	F(THETA(I),PS)
F(THETA(I),PS)		, , , - ,
С	1	U(THETA(I),PS)
U(THETA(I),PF)		
С	2	U(THETA(I),PS)
F(THETA(I), PF)	2	- (
C E (MILEMA (T) DE)	3	F(THETA(I),PS)
F(THETA(I), PF)	4	F(THETA(I),PS)
U(THETA(I), PF)	7	F (IIIEIA(I), IS)
C (IIIIIII(I), (II)		
С	NOTE:	
С	IF THE T	CABLE CALLS FOR BOTH
THE SOLUTION U		
С	AND THE	RIGHT SIDE F AT A
CORNER THEN THE	207.777.701	
C	SOLUTION	N MUST BE SPECIFIED.
C	IDIMF	
C		(OR FIRST) DIMENSION
OF THE ARRAY	11111111000	(OR TINET) BITEROLOR
С	F AS IT	APPEARS IN THE
PROGRAM CALLING		
C	HWSSSP.	THIS PARAMETER IS
USED TO SPECIFY		
C	THE VARI	TABLE DIMENSION OF F.
IDIMF MUST BE	AT LEAST	7 M 1 1
C	AI LEASI	. M+1 •
C		
C ON OUTPUT	F	
С	CONTAINS	THE SOLUTION U(I,J)
OF THE FINITE		
С	DIFFEREN	NCE APPROXIMATION FOR
THE GRID POINT	/ 	-
C MIT AND	('I'HE'I'A (1	I),PHI(J)), I =
1,2,,M+1 AND C	T = 1 2	,N+1 .
C	0 - 1,2,	· · · / IN I I
J		

C	PERTRB
C	IF ONE SPECIFIES A
COMBINATION OF PERIODIC,	
C	DERIVATIVE OR UNSPECIFIED
BOUNDARY	
C	CONDITIONS FOR A POISSON
EQUATION	
C	(LAMBDA = 0), A SOLUTION MAY
NOT EXIST.	
C	PERTRB IS A CONSTANT,
CALCULATED AND	
C	SUBTRACTED FROM F, WHICH
ENSURES THAT A	
C	SOLUTION EXISTS. HWSSSP THEN
COMPUTES	MILE COLUMNON WILLOU TO A
C LEACH COUNTRY	THIS SOLUTION, WHICH IS A
LEAST SQUARES	SOLUTION TO THE ORIGINAL
APPROXIMATION.	SOLUTION TO THE ORIGINAL
C	THIS SOLUTION IS NOT UNIQUE
AND IS	IIIIS SOLUTION IS NOT UNIQUE
C	UNNORMALIZED. THE VALUE OF
PERTRB SHOULD	OMMORTALIZAD. THE VILIOU OF
C	BE SMALL COMPARED TO THE
RIGHT SIDE F.	
C	OTHERWISE , A SOLUTION IS
OBTAINED TO AN	, ii sozozon ze
C	ESSENTIALLY DIFFERENT
PROBLEM. THIS	
C	COMPARISON SHOULD ALWAYS BE
MADE TO INSURE	
C	THAT A MEANINGFUL SOLUTION
HAS BEEN	
C	OBTAINED
C	
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS. EXCEPT FOR
NUMBERS 0 AND 8,	
C	A SOLUTION IS NOT ATTEMPTED.
C	
C	= 0 NO ERROR
C	= 1 TS.LT.0 OR TF.GT.PI

```
= 2 TS.GE.TF
С
                           = 3 MBDCND.LT.1 OR
MBDCND.GT.9
                           = 4 PS.LT.O OR PS.GT.PI+PI
С
                           = 5 PS.GE.PF
С
С
                           = 6 N.LT.5
С
                           = 7 M.LT.5
С
                           = 8 NBDCND.LT.0 OR
NBDCND.GT.4
                           = 9 ELMBDA.GT.0
С
                           = 10 IDIMF.LT.M+1
С
                           = 11 NBDCND EQUALS 1,2 OR 4
AND MBDCND.GE.5
                           = 12 TS.EQ.O AND MBDCND
EQUALS 3,4 OR 8
                           = 13 TF.EQ.PI AND MBDCND
EQUALS 2,3 OR 6
                           = 14 MBDCND EQUALS 5,6 OR 9
AND TS.NE.0
                           = 15 MBDCND.GE.7 AND TF.NE.PI
С
C
                           = 20 If the dynamic
allocation of real and
                                complex work space
required for solution
                                fails (for example if
N,M are too large
                                for your computer)
С
С
C SPECIAL CONDITIONS
                        NONE
С
C I/O
                        NONE
С
C PRECISION
                         SINGLE
C REQUIRED files
fish.f,genbun.f,gnbnaux.f,comf.f
C LANGUAGE
                        FORTRAN 90
C
C HISTORY
                        WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE
                         1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE
```

С	LIBRARIES IN JANUARY 1980.
C	Revised in June 2004 by John
Adams using	
C	Fortran 90 dynamically
allocated work space.	
C	
C PORTABILITY	FORTRAN 90
С	
C ALGORITHM	THE ROUTINE DEFINES THE FINITE
DIFFERENCE	
С	EQUATIONS, INCORPORATES
BOUNDARY DATA, AND	
С	ADJUSTS THE RIGHT SIDE OF
SINGULAR SYSTEMS	
С	AND THEN CALLS GENBUN TO SOLVE
THE SYSTEM.	
С	
C TIMING	FOR LARGE M AND N, THE
OPERATION COUNT	
С	IS ROUGHLY PROPORTIONAL TO
С	M*N* (LOG2 (N)
С	BUT ALSO DEPENDS ON INPUT
PARAMETERS NBDCND	
C	AND MBDCND.
С	
C ACCURACY	THE SOLUTION PROCESS EMPLOYED
RESULTS IN A LOSS	
C	OF NO MORE THAN THREE
SIGNIFICANT DIGITS FOR N	
C	AND M AS LARGE AS 64. MORE
DETAILS ABOUT	
C	ACCURACY CAN BE FOUND IN THE
DOCUMENTATION FOR	
C	SUBROUTINE GENBUN WHICH IS THE
ROUTINE THAT	
C	SOLVES THE FINITE DIFFERENCE
EQUATIONS.	
С	
C REFERENCES	P. N. SWARZTRAUBER, "THE DIRECT
SOLUTION OF	
C	THE DISCRETE POISSON EQUATION
ON THE SURFACE OF	
C	A SPHERE", S.I.A.M. J. NUMER.
ANAL.,15(1974),	

POIS3D

```
file pois3d.txt (documentation for the FISHPACK
С
solver POIS3D)
С
С
                         copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
С
                             all rights reserved
С
*
С
                           FISHPACK90 version 1.1
*
С
```

```
A Package of Fortran 77 and 90
С
                        Subroutines and Example Programs
C
С
                       for Modeling Geophysical Processes
C
C
С
                                      by
               John Adams, Paul Swarztrauber and Roland
С
Sweet
\mathsf{C}
С
                                      of
С
                 the National Center for Atmospheric
Research
                        Boulder, Colorado (80307)
U.S.A.
С
                           which is sponsored by
С
С
C
                      the National Science Foundation
С
С
      SUBROUTINE POIS3D
```

```
(LPEROD, L, C1, MPEROD, M, C2, NPEROD, N, A, B, C, LDIMF,
С
                          MDIMF, F, IERROR)
C
C
C DIMENSION OF
                          A(N), B(N), C(N),
F(LDIMF, MDIMF, N)
C ARGUMENTS
C LATEST REVISION
                         June 2004
C PURPOSE
                          SOLVES THE LINEAR SYSTEM OF
EQUATIONS
                          FOR UNKNOWN X VALUES, WHERE
I=1,2,...,L,
                          J=1,2,...,M, AND K=1,2,...,N
С
                          C1*(X(I-1,J,K) -2.*X(I,J,K)
+X(I+1,J,K)) +
                          C2*(X(I,J-1,K) -2.*X(I,J,K)
+X(I,J+1,K)) +
                          A(K) *X(I,J,K-1) +B(K) *X(I,J,K) +
C(K) *X(I,J,K+1)
                          = F(I,J,K)
C
С
С
                          THE INDICES K-1 AND K+1 ARE
EVALUATED MODULO N,
                          I.E. X(I,J,0)=X(I,J,N) AND
X(I,J,N+1) = X(I,J,1).
                          THE UNKNOWNS
                          X(0,J,K), X(L+1,J,K), X(I,0,K),
AND X(I,M+1,K)
                          ARE ASSUMED TO TAKE ON CERTAIN
PRESCRIBED
                          VALUES DESCRIBED BELOW.
C
C USAGE
                          CALL POIS3D
(LPEROD, L, C1, MPEROD, M, C2, NPEROD,
C
                          N, A, B, C, LDIMF, MDIMF, F, IERROR)
С
C ARGUMENTS
C ON INPUT
С
                          LPEROD
C
                             INDICATES THE VALUES THAT
```

```
X(0,J,K) AND
                                X(L+1,J,K) ARE ASSUMED TO
HAVE.
                                = 0 X(0,J,K) = X(L,J,K),
X(L+1, J, K) = X(1, J, K)
                                = 1 \times (0, J, K) = 0,
X(L+1,J,K) = 0
                                = 2 \times (0, J, K) = 0,
X(L+1,J,K) = X(L-1,J,K)
                                = 3 \times (0, J, K) = X(2, J, K),
X(L+1, J, K) = X(L-1, J, K)
                                = 4 \times (0, J, K) = X(2, J, K),
X(L+1, J, K) = 0.
C
С
                              L
С
                                THE NUMBER OF UNKNOWNS IN THE
I-DIRECTION.
                                L MUST BE AT LEAST 3.
C
\mathsf{C}
С
                              C1
С
                                REAL CONSTANT IN THE ABOVE
LINEAR SYSTEM
                                OF EQUATIONS TO BE SOLVED.
C
С
С
                              MPEROD
С
                                INDICATES THE VALUES THAT
X(I,0,K) AND
C
                                X(I,M+1,K) ARE ASSUMED TO
HAVE.
                                = 0 X(I, 0, K) = X(I, M, K),
X(I,M+1,K) = X(I,1,K)
                                = 1 \times (I, 0, K) = 0,
X(I,M+1,K) = 0
                                = 2 \times (I, 0, K) = 0,
X(I,M+1,K) = X(I,M-1,K)
                                = 3 \times (I, 0, K) = X(I, 2, K)
X(I,M+1,K) = X(I,M-1,K)
                                = 4 \times (I, 0, K) = X(I, 2, K)
X(I,M+1,K)=0
С
C
С
                                THE NUMBER OF UNKNOWNS IN THE
J-DIRECTION.
                                M MUST BE AT LEAST 3.
```

```
С
С
                          C2
C
                            REAL CONSTANT IN THE ABOVE
LINEAR SYSTEM
                            OF EQUATIONS TO BE SOLVED.
С
С
                         NPEROD
С
                            = 0 IF A(1) AND C(N) ARE NOT
ZERO.
                            = 1 	ext{ IF } A(1) = C(N) = 0.
C
С
С
                         Ν
С
                            THE NUMBER OF UNKNOWNS IN THE
K-DIRECTION.
                            N MUST BE AT LEAST 3.
С
С
                         A, B, C
С
                            ONE-DIMENSIONAL ARRAYS OF
LENGTH N THAT
                            SPECIFY THE COEFFICIENTS IN
THE LINEAR
                            EQUATIONS GIVEN ABOVE.
С
                            IF NPEROD = 0 THE ARRAY
ELEMENTS MUST NOT
C
                            DEPEND UPON INDEX K, BUT MUST
BE CONSTANT.
                            SPECIFICALLY, THE SUBROUTINE
CHECKS THE
С
                            FOLLOWING CONDITION
С
                             A(K) = C(1)
С
                             C(K) = C(1)
С
                             B(K) = B(1)
С
                            FOR K=1, 2, ..., N.
С
С
                          LDIMF
С
                            THE ROW (OR FIRST) DIMENSION
OF THE THREE-
                            DIMENSIONAL ARRAY F AS IT
APPEARS IN THE
                            PROGRAM CALLING POIS3D. THIS
PARAMETER IS
                            USED TO SPECIFY THE VARIABLE
DIMENSION
```

1	
C	OF F. LDIMF MUST BE AT LEAST
L.	
C	
C	MDIMF
C	THE COLUMN (OR SECOND)
DIMENSION OF THE THREE	THE COLORN (OR BECOME)
C	DIMENSIONAL ARRAY F AS IT
APPEARS IN THE	
C	PROGRAM CALLING POIS3D. THIS
PARAMETER IS	
C	USED TO SPECIFY THE VARIABLE
DIMENSION	
C	OF F. MDIMF MUST BE AT LEAST
M.	
C	
C	F
С	A THREE-DIMENSIONAL ARRAY
THAT SPECIFIES THE	
C	VALUES OF THE RIGHT SIDE OF
	VALUES OF THE KIGHT SIDE OF
THE LINEAR SYSTEM	
C	OF EQUATIONS GIVEN ABOVE. F
MUST BE	
C	DIMENSIONED AT LEAST L X M X
N.	
С	
C ON OUTPUT	
C	
C	F
C	CONTAINS THE SOLUTION X.
C	
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C C	PARAMETERS. EXCEPT FOR
	FARAMETERS. EXCEPT FOR
NUMBER ZERO, A	
C	SOLUTION IS NOT ATTEMPTED.
C	= 0 NO ERROR
C	= 1 IF LPEROD .LT. 0 OR .GT.
4	
C	= 2 IF L .LT. 3
C	= 3 IF MPEROD .LT. 0 OR .GT.
	- 3 II MELNOU .LII. U OK .GI.
4	
С	= 4 IF M .LT. 3
C	= 5 IF NPEROD .LT. 0 OR .GT.

```
С
                          = 6 IF N .LT. 3
                          = 7 IF LDIMF .LT. L
С
С
                          = 8 IF MDIMF .LT. M
С
                           = 9 IF A(K) .NE. C(1) OR
C(K) .NE. C(1)
                               OR B(I) .NE.B(1) FOR
SOME K=1,2,...,N.
                          = 10 IF NPEROD = 1 AND A(1)
.NE. 0
С
                               OR C(N) .NE. 0
С
                          = 20 If the dynamic
allocation of real and
                               complex work space
required for solution
                               fails (for example if
N,M are too large
                                for your computer)
С
                          SINCE THIS IS THE ONLY MEANS
OF INDICATING A
                          POSSIBLY INCORRECT CALL TO
POIS3D, THE USER
                          SHOULD TEST IERROR AFTER THE
CALL.
C SPECIAL CONDITIONS
                       NONE
C I/O
                       NONE
C PRECISION
                        SINGLE
C REQUIRED files fish.f, comf.f, fftpack.f
                        FORTRAN 90
C LANGUAGE
С
C HISTORY
                       WRITTEN BY ROLAND SWEET AT NCAR
IN THE LATE
                        1970'S. RELEASED ON NCAR'S
PUBLIC SOFTWARE
                        LIBRARIES IN JANUARY, 1980.
                        Revised in June 2004 by John
Adams using
                        Fortran 90 dynamically
```

allocated work space.	
C	
C PORTABILITY	FORTRAN 90
C	
C ALGORITHM	THIS SUBROUTINE SOLVES THREE-
DIMENSIONAL BLOCK	
C	TRIDIAGONAL LINEAR SYSTEMS
ARISING FROM FINITE	
C	DIFFERENCE APPROXIMATIONS TO
THREE-DIMENSIONAL	DITTELLUCE ATTICALITY TO TO
C C	DOIGGON FOLLATIONS LISTED THE FET
	POISSON EQUATIONS USING THE FFT
PACKAGE	
C	FFTPACK WRITTEN BY PAUL
SWARZTRAUBER.	
C	
C TIMING	FOR LARGE L, M AND N, THE
OPERATION COUNT	
C	IS ROUGHLY PROPORTIONAL TO
C	L*M*N* (LOG2 (L) + LOG2 (M) +5)
C	BUT ALSO DEPENDS ON INPUT
PARAMETERS LPEROD	
C	AND MPEROD.
С	
C ACCURACY	TO MEASURE THE ACCURACY OF THE
ALGORITHM A	
С	UNIFORM RANDOM NUMBER GENERATOR
WAS USED TO	
С	CREATE A SOLUTION ARRAY X FOR
THE SYSTEM GIVEN	
C	IN THE 'PURPOSE' SECTION WITH
C	A(K) = C(K) = -0.5*B(K) = 1,
K=1,2,,N	A(R) = C(R) = 0.0 B(R) = 1,
C	AND, WHEN NPEROD = 1
C	A(1) = C(N) = 0
C	A(N) = C(1) = 2.
C	
C	THE SOLUTION X WAS SUBSTITUTED
INTO THE GIVEN	
C	SYSTEM AND, USING DOUBLE
PRECISION, A RIGHT	
C	SIDE Y WAS COMPUTED. USING
THIS ARRAY Y	
C	SUBROUTINE POIS3D WAS CALLED TO
PRODUCE AN	

C RELATIVE ERROR	APPROXIMA	ATE SOLUTIO	ON Z.
C C X(I,J,K)))/MAX(ABS(X(I,J	•	ABS(Z(I,J,	K) –
C C MAXIMA ARE TAKEN	WAS COMPU	JTED, WHER	E THE TWO
C AND K=1,2,,N.	OVER I=1,	,2,,L,	J=1,2,,M
C TABLE BELOW FOR			VEN IN THE
C N. C	SOME TYPI	ICAL VALUE	S OF L,M AND
C C E	L(=M=N)	LPEROD	MPEROD
C 			
C C 1.E-13	16	0	0
C 4.E-13	15	1	1
C 2.E-13	17	3	3
C 2.E-13 C	32 31	0	0
2.E-12 C	33	3	3
7.E-13 C	33	J	3
C REFERENCES	NONE		
********** *****	*****	*****	*****

```
С
      file poistg.txt (documentation for the FISHPACK
С
solver POISTG)
С
                         copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
С
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
С
                      for Modeling Geophysical Processes
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
```

```
С
                                    of
С
С
                the National Center for Atmospheric
Research
                      Boulder, Colorado (80307)
U.S.A.
C
С
                          which is sponsored by
С
                    the National Science Foundation
С
*
С
     *
      SUBROUTINE POISTG
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y, IERROR)
С
C DIMENSION OF
                        A(M), B(M), C(M), Y(IDIMY, N)
C ARGUMENTS
C LATEST REVISION June 2004
С
C PURPOSE
                        SOLVES THE LINEAR SYSTEM OF
EQUATIONS
С
                        FOR UNKNOWN X VALUES, WHERE
I=1,2,...,M
                         AND J=1,2,...,N
C
С
                         A(I) *X(I-1,J) + B(I) *X(I,J) +
C(I) *X(I+1,J)
                         + X(I,J-1) - 2.*X(I,J) +
X(I,J+1)
                         = Y(I,J)
```

```
С
                             THE INDICES I+1 AND I-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 X(I,1)
OR -X(I,1), AND
                             X(I,N+1) MAY BE EQUAL TO X(I,N)
OR - X(I, N),
                             DEPENDING ON AN INPUT
PARAMETER.
C
C USAGE
                             CALL POISTG
(NPEROD, N, MPEROD, M, A, B, C, IDIMY, Y,
                                             IERROR)
С
C ARGUMENTS
C ON INPUT
С
С
                             NPEROD
C
                                INDICATES VALUES WHICH X(I,0)
AND X(I, N+1)
                                ARE ASSUMED TO HAVE.
C
С
                                = 1 \text{ IF } X(I,0) = -X(I,1) \text{ AND}
X(I,N+1) = -X(I,N)
                                = 2 \text{ IF } X(I,0) = -X(I,1) \text{ AND}
X(I,N+1) = X(I,N)
                                = 3 \text{ IF } X(I,0) = X(I,1) \text{ AND}
X(I,N+1) = X(I,N)
                                = 4 \text{ IF } X(I,0) = X(I,1) \text{ AND}
X(I,N+1) = -X(I,N)
С
С
                             Ν
С
                                THE NUMBER OF UNKNOWNS IN THE
J-DIRECTION.
С
                                N MUST BE GREATER THAN 2.
С
С
                             MPEROD
С
                                = 0 \text{ IF A (1)} \text{ AND C (M)} \text{ ARE NOT}
ZERO
С
                               = 1 \text{ IF A(1)} = C(M) = 0
С
С
```

```
THE NUMBER OF UNKNOWNS IN THE
I-DIRECTION.
С
                           M MUST BE GREATER THAN 2.
С
С
                        A,B,C
                           ONE-DIMENSIONAL ARRAYS OF
LENGTH M THAT
                           SPECIFY THE COEFFICIENTS IN
THE LINEAR
                          EQUATIONS GIVEN ABOVE. IF
MPEROD = 0 THE
                          ARRAY ELEMENTS MUST NOT
DEPEND ON INDEX I,
                           BUT MUST BE CONSTANT.
SPECIFICALLY, THE
                           SUBROUTINE CHECKS THE
FOLLOWING CONDITION
                            A(I) = C(1)
С
                             B(I) = B(1)
С
                             C(I) = C(1)
С
                           FOR I = 1, 2, ..., M.
С
С
                         IDIMY
                           THE ROW (OR FIRST) DIMENSION
OF THE TWO-
                           DIMENSIONAL ARRAY Y AS IT
APPEARS IN THE
                           PROGRAM CALLING POISTG. THIS
PARAMETER IS
                           USED TO SPECIFY THE VARIABLE
DIMENSION OF Y.
                           IDIMY MUST BE AT LEAST M.
C
С
C
                           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 X N.
C ON OUTPUT
```

C	Υ
С	CONTAINS THE SOLUTION X.
С	
С	IERROR
С	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS. EXCEPT FOR
NUMBER ZERO, A	
C	SOLUTION IS NOT ATTEMPTED.
C	= 0 NO ERROR
С	= 1 IF M .LE. 2
С	= 2 IF N .LE. 2
C	= 3 IDIMY .LT. M
С	= 4 IF NPEROD .LT. 1 OR
NPEROD .GT. 4	
С	= 5 IF MPEROD .LT. 0 OR
MPEROD .GT. 1	
С	= 6 IF MPEROD $= 0$ AND A(I)
.NE. C(1)	, ,
C	OR $B(I)$.NE. $B(1)$ OR
C(I) .NE. C(1)	
C (2, 31.23 3 (2,	FOR SOME $I = 1, 2, \ldots$
M.	1010 20111 1 1, 2,,
C	= 7 IF MPEROD .EQ. 1 .AND.
C	(A(1).NE.0 .OR.
C(M).NE.0)	(A(I).NE.O .OK.
C (FI) . NE. O)	= 20 If the dynamic
allocation of real and	- 20 II the dynamic
C C	complex work space
required for solution	complex work space
C C	fails (for example if
	fails (for example if
N,M are too large	<i>5</i>
C	for your computer)
C	
C	SINCE THIS IS THE ONLY MEANS
OF INDICATING A	
С	POSSIBLY INCORRECT CALL TO
POIS3D, THE USER	
C	SHOULD TEST IERROR AFTER THE
CALL.	
C	
С	
С	
C I/O	NONE

C DDECTCION	CINCLE
C PRECISION	SINGLE
C REQUIRED files	fish.f,gnbnaux.f,comf.f
C LANGUAGE	FORTRAN 90
C HISTORY IN THE LATE	WRITTEN BY ROLAND SWEET AT NCAR
C PUBLIC SOFTWARE	1970's. RELEASED ON NCAR'S
C	LIBRARIES IN JANUARY, 1980. Revised in June 2004 by John
Adams using c allocated work space.	Fortran 90 dynamically
C C PORTABILITY	FORTRAN 90
C ALGORITHM IMPLEMENTATION OF THE	THIS SUBROUTINE IS AN
C REFERENCE BELOW.	ALGORITHM PRESENTED IN THE
C TIMING EXECUTION TIME IS	FOR LARGE M AND N, THE
C M*N*LOG2(N).	ROUGHLY PROPORTIONAL TO
C ACCURACY ALGORITHM A	TO MEASURE THE ACCURACY OF THE
C WAS USED TO	UNIFORM RANDOM NUMBER GENERATOR
C THE SYSTEM GIVEN	CREATE A SOLUTION ARRAY X FOR
C WITH	IN THE 'PURPOSE' SECTION ABOVE,
C I=1,2,,M	A(I) = C(I) = -0.5*B(I) = 1,
C C C	AND, WHEN MPEROD = 1 A(1) = C(M) = 0 B(1) = B(M) = -1.
C	THE SOLUTION X WAS SUBSTITUTED

INTO THE GIVEN					
C	SYSTEM AND, USING DOUBLE				
PRECISION, A RIGHT SID					
C	Y WAS CC	MPUTED. US	SING THIS		
ARRAY Y SUBROUTINE					
C	POISTG W	AS CALLED 7	O PRODUCE AN		
APPROXIMATE					
C	SOLUTION Z. THEN THE RELATIVE				
ERROR, DEFINED A					
C	E = MA	X(ABS(Z(I,	J) —		
X(I,J))/MAX(ABS(X(I,J)))					
C		E TWO MAXIN	IA ARE TAKEN		
OVER I=1,2,,M	***************************************	1			
C	AND J=1,2,,N, WAS COMPUTED.				
VALUES OF E ARE					
C	GTVEN IN	THE TABLE	BELOW FOR		
SOME TYPICAL VALUE	017217 117		DLLOW 1 OIX		
C	OF M AND N.				
C	0_ 11 1_12				
C	M (=N)	MPEROD	NPEROD		
E	11 (11)	111 111 (0.1)	111 2110 2		
C					
C					
C	31	0-1	1-4		
9.E-13	91	0 1	<u> </u>		
C C	31	1	1		
4.E-13	91	_	_		
C	31	1	3		
3.E-13	01	_	o l		
C C	32	0-1	1-4		
3.E-12	<i>52</i>	V 1			
C C	32	1	1		
3.E-13	02	_	-		
C C	32	1	3		
1.E-13	<i>52</i>	<u> </u>	J		
C C	33	0-1	1-4		
1.E-12	55	O T	± 1		
C C	33	1	1		
4.E-13	33	_	-		
C C	33	1	3		
1.E-13		<u> </u>	J		
C C	63	0-1	1-4		
3.E-12	0.5	O I	T 1		
J.H 14					

C	63	1	1
1.E-12 C	63	1	3
2.E-13 C	64	0-1	1-4
4.E-12	-		
C 1.E-12	64	1	1
C 6.E-13	64	1	3
C	65	0-1	1-4
2.E-13 C	65	1	1
1.E-11	CF	1	3
C 4.E-13	65	1	3
C C REFERENCES	SCHUMANN, U	AND R. S	WEET."A
DIRECT METHOD	·		·
C EQUATION WITH	FOR THE SOL	UTION OF P	OISSON"S
C A STAGGERED	NEUMANN BOU	NDARY COND	ITIONS ON
С	GRID OF ARB	ITRARY SIZ	E," J.
COMP. PHYS.	20(1976), P	P. 171-182	
C			

SEPELI

```
copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
С
Research
С
                              all rights reserved
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
С
                the National Center for Atmospheric
Research
```

```
С
C
                        Boulder, Colorado (80307)
U.S.A.
С
C
                           which is sponsored by
С
С
                     the National Science Foundation
C
С
      SUBROUTINE SEPELI
(INTL, IORDER, A, B, M, MBDCND, BDA, ALPHA, BDB, BETA, C,
D, N, NBDCND, BDC, GAMA, BDD, XNU, COFX, COFY, GRHS,
                         USOL, IDMN, W, PERTRB, IERROR)
С
C DIMENSION OF
                         BDA(N+1), BDB(N+1), BDC(M+1),
BDD(M+1),
C ARGUMENTS
                         USOL (IDMN, N+1), GRHS (IDMN, N+1),
                   JUNE 2004
C LATEST REVISION
С
C PURPOSE
                         SEPELI SOLVES FOR EITHER THE
SECOND-ORDER
                          FINITE DIFFERENCE APPROXIMATION
С
OR A
                          FOURTH-ORDER APPROXIMATION TO A
SEPARABLE
С
                          ELLIPTIC EQUATION
С
С
                           AF(X)*DU/DX + BF(X)*DU/DX +
C
CF(X)*U+
С
                                       2
C
                            DF(Y)*DU/DY + EF(Y)*DU/DY +
FF(Y)*U
```

```
C
С
                           = G(X,Y)
С
С
                          ON A RECTANGLE (X GREATER THAN
OR EQUAL TO A
                         AND LESS THAN OR EQUAL TO B; Y
GREATER THAN
                         OR EQUAL TO C AND LESS THAN OR
EQUAL TO D).
                         ANY COMBINATION OF PERIODIC OR
MIXED BOUNDARY
                          CONDITIONS IS ALLOWED.
C
С
                          THE POSSIBLE BOUNDARY
CONDITIONS ARE:
C
                          IN THE X-DIRECTION:
                          (0) PERIODIC, U(X+B-A, Y)=U(X, Y)
FOR ALL
                             Y,X (1) U(A,Y), U(B,Y) ARE
SPECIFIED FOR
С
                              ALL Y
C
                          (2) U(A, Y),
DU(B,Y)/DX+BETA*U(B,Y) ARE
                              SPECIFIED FOR ALL Y
С
С
                          (3)
DU(A,Y)/DX+ALPHA*U(A,Y),DU(B,Y)/DX+
                              BETA*U(B,Y) ARE SPECIFIED
FOR ALL Y
                          (4)
DU(A,Y)/DX+ALPHA*U(A,Y),U(B,Y) ARE
C
                              SPECIFIED FOR ALL Y
С
С
                          IN THE Y-DIRECTION:
                          (0) PERIODIC, U(X,Y+D-C)=U(X,Y)
FOR ALL X, Y
                          (1) U(X,C),U(X,D) ARE SPECIFIED
C
FOR ALL X
                          (2)
U(X,C), DU(X,D)/DY+XNU*U(X,D) ARE
С
                              SPECIFIED FOR ALL X
                          (3)
DU(X,C)/DY+GAMA*U(X,C),DU(X,D)/DY+
C
                              XNU*U(X,D) ARE SPECIFIED
FOR ALL X
```

```
DU(X,C)/DY+GAMA*U(X,C),U(X,D) ARE
С
                               SPECIFIED FOR ALL X
C
C USAGE
                           CALL SEPELI
(INTL, IORDER, A, B, M, MBDCND, BDA,
ALPHA, BDB, BETA, C, D, N, NBDCND, BDC,
GAMA, BDD, XNU, COFX, COFY, GRHS, USOL,
IDMN, W, PERTRB, IERROR)
C ARGUMENTS
C ON INPUT
                          INTL
                            = 0 ON INITIAL ENTRY TO
SEPELI OR IF ANY
                                 OF THE ARGUMENTS C, D, N,
NBDCND, COFY
                                 ARE CHANGED FROM A
PREVIOUS CALL
                            = 1 IF C, D, N, NBDCND, COFY
ARE UNCHANGED
                                 FROM THE PREVIOUS CALL.
С
С
                           IORDER
                            = 2 IF A SECOND-ORDER
APPROXIMATION
                                 IS SOUGHT
                            = 4 IF A FOURTH-ORDER
APPROXIMATION
                                 IS SOUGHT
С
С
                          A,B
С
                             THE RANGE OF THE X-
INDEPENDENT VARIABLE,
                            I.E., X IS GREATER THAN OR
EQUAL TO A
                            AND LESS THAN OR EQUAL TO B.
A MUST BE
                            LESS THAN B.
C
С
\mathsf{C}
                          M
                             THE NUMBER OF PANELS INTO
```

WHICH THE	
C	INTERVAL [A,B] IS SUBDIVIDED.
HENCE,	
C	THERE WILL BE M+1 GRID POINTS
IN THE X-	
С	DIRECTION GIVEN BY XI=A+(I-
1) *DLX	`
C	FOR $I=1,2,\ldots,M+1$ WHERE
DLX=(B-A)/M IS	101(11/2//11/1 WILLIA
C C	THE PANEL WIDTH. M MUST BE
LESS THAN	THE TANKE WIDTH. IN MOST BE
	TOWN AND COEAMED MILAN E
C	IDMN AND GREATER THAN 5.
C	
C	MBDCND
C	INDICATES THE TYPE OF
BOUNDARY CONDITION	
C	AT X=A AND X=B
C	
C	= 0 IF THE SOLUTION IS
PERIODIC IN X, I.E.,	
C	U(X+B-A,Y)=U(X,Y) FOR
ALL Y,X	
C	= 1 IF THE SOLUTION IS
SPECIFIED AT X=A	
C	AND X=B, I.E., U(A,Y) AND
U(B,Y) ARE	1110 11 D, 1.1., 0 (11, 1) 1110
C C	SPECIFIED FOR ALL Y
C	= 2 IF THE SOLUTION IS
	- Z IF THE SOLUTION IS
SPECIFIED AT X=A AND	THE DOLLING TO CONDITION TO
C	THE BOUNDARY CONDITION IS
MIXED AT X=B,	
C	I.E., U(A,Y) AND
DU(B,Y)/DX+BETA*U(B,Y)	
C	ARE SPECIFIED FOR ALL Y
C	= 3 IF THE BOUNDARY
CONDITIONS AT X=A AND	
С	X=B ARE MIXED, I.E.,
С	DU(A,Y)/DX+ALPHA*U(A,Y)
AND	
C	DU(B,Y)/DX+BETA*U(B,Y)
ARE SPECIFIED	- () - /
C	FOR ALL Y
C	= 4 IF THE BOUNDARY CONDITION
AT X=A IS	I II IIII DOUDIII(I CONDIIION
VI V_V IN	

```
MIXED AND THE SOLUTION IS
SPECIFIED
                               AT X=B, I.E.,
DU(A,Y)/DX+ALPHA*U(A,Y)
                               AND U(B,Y) ARE SPECIFIED
FOR ALL Y
C
С
                         BDA
                           A ONE-DIMENSIONAL ARRAY OF
С
LENGTH N+1
                           THAT SPECIFIES THE VALUES OF
С
С
                           DU(A,Y)/DX+ ALPHA*U(A,Y) AT
X=A, WHEN
С
                           MBDCND=3 OR 4.
С
                           BDA(J) =
DU(A,YJ)/DX+ALPHA*U(A,YJ),
                           J=1,2,...,N+1. WHEN MBDCND
HAS ANY OTHER
                           OTHER VALUE, BDA IS A DUMMY
PARAMETER.
С
C
                        ALPHA
C
                           THE SCALAR MULTIPLYING THE
SOLUTION IN
                           CASE OF A MIXED BOUNDARY
CONDITION AT X=A
                           (SEE ARGUMENT BDA). IF
MBDCND IS NOT
                           EQUAL TO 3 OR 4 THEN ALPHA IS
A DUMMY
С
                           PARAMETER.
С
С
                         BDB
                           A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1
                           THAT SPECIFIES THE VALUES OF
С
                           DU(B,Y)/DX+ BETA*U(B,Y) AT
С
X=B.
С
                           WHEN MBDCND=2 OR 3
                           BDB(J) =
DU(B,YJ)/DX+BETA*U(B,YJ),
                           J=1,2,...,N+1. WHEN MBDCND
HAS ANY OTHER
                           OTHER VALUE, BDB IS A DUMMY
```

PARAMETER.	
C	
C	BETA
C	THE SCALAR MULTIPLYING THE
SOLUTION IN	
C	CASE OF A MIXED BOUNDARY
CONDITION AT	CASE OF A PILADE BOONDARI
	V D (CDD ADCIMUM DDD) TO
C	X=B (SEE ARGUMENT BDB). IF
MBDCND IS	
C	NOT EQUAL TO 2 OR 3 THEN BETA
IS A DUMMY	
C	PARAMETER.
C	
C	C, D
C	THE RANGE OF THE Y-
INDEPENDENT VARIABLE,	
C	I.E., Y IS GREATER THAN OR
EQUAL TO C	T.E., I IO ORDITER THE OR
C	AND LESS THAN OR EQUAL TO D.
C MUST BE	AND DESCRIPTION ON DOOR TO D.
	I DOG BUAN D
C	LESS THAN D.
С	
C	N
C	THE NUMBER OF PANELS INTO
WHICH THE	
C	INTERVAL [C,D] IS SUBDIVIDED.
C	HENCE, THERE WILL BE N+1 GRID
POINTS	
C	IN THE Y-DIRECTION GIVEN BY
C	YJ=C+(J-1)*DLY FOR
J=1,2,,N+1 WHERE	
C C	DLY=(D-C)/N IS THE PANEL
WIDTH.	PHI (P C)/IN IO IIID LUMBII
	TNI ADDIMIONI NI MIOM DE
C CDEATED THAN 4	IN ADDITION, N MUST BE
GREATER THAN 4.	
С	
С	NBDCND
C	INDICATES THE TYPES OF
BOUNDARY CONDITIONS	
С	AT Y=C AND Y=D
С	
С	= 0 IF THE SOLUTION IS
PERIODIC IN Y,	
C	I.E., $U(X,Y+D-C)=U(X,Y)$

FOR ALL X,	Y	= 1	IF THE SOLUTION IS
SPECIFIED C	AT Y=C		AND $Y = D$, I.E., $U(X,C)$
AND U(X,D)			ARE SPECIFIED FOR ALL X
C SPECIFIED	AT Y=C	= 2	IF THE SOLUTION IS
C CONDITION	TS MTXED		AND THE BOUNDARY
C	10 111111111111111111111111111111111111		AT Y=D, I.E., U(X,C) AND DU(X,D)/DY+XNU*U(X,D) ARE
SPECIFIED C			FOR ALL X
C CONDITIONS	ARE MIXED	= 3	IF THE BOUNDARY
C C			AT Y=C AND Y=D, I.E., DU(X,D)/DY+GAMA*U(X,C)
AND C			DU(X,D)/DY+XNU*U(X,D) ARE
SPECIFIED C			FOR ALL X
C IS MIXED		= 4	IF THE BOUNDARY CONDITION
C IS SPECIFI	ED		AT Y=C AND THE SOLUTION
C DU(X,C)/DY	+GAMA*U(X,C)		AT Y=D, I.E.
C FOR ALL X			AND U(X,D) ARE SPECIFIED
C		BDC	
C LENGTH M+1			NE-DIMENSIONAL ARRAY OF
C			I SPECIFIES THE VALUE OF X,C)/DY+GAMA*U(X,C) AT
Y=C.		WHE	N NBDCND=3 OR 4 BDC(I) =
DU(XI,C)/DC	Y +		A*U(XI,C), I=1,2,,M+1.
C VALUE, BDC			N NBDCND HAS ANY OTHER
C		IS 7	A DUMMY PARAMETER.

C	GAMA
C	THE SCALAR MULTIPLYING THE
SOLUTION IN	
C	CASE OF A MIXED BOUNDARY
CONDITION AT	
C	Y=C (SEE ARGUMENT BDC). IF
NBDCND IS	I-C (SEE ARGOMENI BDC). IF
	NOT TOUR TO 2 OD 4 HITEL CANA
C	NOT EQUAL TO 3 OR 4 THEN GAMA
IS A DUMMY	
C	PARAMETER.
C	
C	BDD
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1	
C	THAT SPECIFIES THE VALUE OF
C	DU(X,D)/DY + XNU*U(X,D) AT
Y=C.	
С	WHEN NBDCND=2 OR 3 BDD(I) =
DU(XI,D)/DY +	
C	XNU*U(XI,D), I=1,2,,M+1.
C	WHEN NBDCND HAS ANY OTHER
	WHEN INDUCTED HAS ANT OTHER
VALUE, BDD	
C	IS A DUMMY PARAMETER.
C	
С	XNU
C	THE SCALAR MULTIPLYING THE
SOLUTION IN	
C	CASE OF A MIXED BOUNDARY
CONDITION AT	
C	Y=D (SEE ARGUMENT BDD). IF
NBDCND IS	
C	NOT EQUAL TO 2 OR 3 THEN XNU
IS A	
С	DUMMY PARAMETER.
C	
C	COFX
C	A USER-SUPPLIED SUBPROGRAM
WITH	A OSEN-SOFFLIED SOBFROGRAM
C	PARAMETERS X, AFUN, BFUN,
CFUN WHICH	
C	RETURNS THE VALUES OF THE X-
DEPENDENT	
C	COEFFICIENTS AF(X), BF(X),
CF(X) IN THE	

C	ELLIPTIC EQUATION AT X.
C	
C	COFY
С	A USER-SUPPLIED SUBPROGRAM
WITH PARAMETERS	
C	Y, DFUN, EFUN, FFUN WHICH
RETURNS THE	
С	VALUES OF THE Y-DEPENDENT
COEFFICIENTS	
С	DF(Y), $EF(Y)$, $FF(Y)$ IN THE
ELLIPTIC	
C	EQUATION AT Y.
C	_
C	NOTE: COFX AND COFY MUST BE
DECLARED	
C	EXTERNAL IN THE CALLING
ROUTINE.	
C	THE VALUES RETURNED IN AFUN
AND DFUN	
C	MUST SATISFY AFUN*DFUN
GREATER THAN 0	
C	FOR A LESS THAN X LESS THAN
B, C LESS	
C	THAN Y LESS THAN D (SEE
IERROR=10).	`
C	THE COEFFICIENTS PROVIDED MAY
LEAD TO A	
C	MATRIX EQUATION WHICH IS NOT
DIAGONALLY	~
C	DOMINANT IN WHICH CASE
SOLUTION MAY FAIL	
C	(SEE IERROR=4).
C	,
C	GRHS
C	A TWO-DIMENSIONAL ARRAY THAT
SPECIFIES THE	
C	VALUES OF THE RIGHT-HAND SIDE
OF THE	
C	ELLIPTIC EQUATION, I.E.,
C	GRHS $(I,J) = G(XI,YI)$, FOR
I=2,,M,	- (, - , - , - , - , , - , - , , - , -
C	J=2,,N. AT THE
BOUNDARIES, GRHS IS	, ,
C	DEFINED BY
<u> </u>	

C			
C	MDDCND	CDUC (1 T)	
	MDDCND	GRHS (1, J)	
GRHS (M+1,J)			
C	0	$C(\Lambda, VT)$	C (D VI)
C	1	G(A,YJ) *	G(B , YJ) *
C	2	*	
J=1,2,,N+1	2	^	G(B,YJ)
C C	3	G(A,YJ)	C(P VT)
C	4	G(A, YJ)	*
C	4	G(A, 10)	
C	NBDCND	GRHS(I,1)	
GRHS(I,N+1)	NDDCND	GIMID(I,I)	
C C			
C	0	G(XI,C)	G(XT ₋ D)
C	1	*	*
C	2	*	G(XI,D)
I=1,2,,M+1	2		O (211 / D)
C	3	G(XI,C)	G(XT.D)
C	4	G(XI,C)	*
C	1	0 (211/0)	
C	WHERE *	MEANS THESE	
QUANTITIES ARE NOT USED.	***************************************		
C	GRHS SHO	OULD BE DIME	NSTONED
IDMN BY AT LEAST	01410 011	3022 22 222	
C	N+1 IN 5	THE CALLING 1	ROUTINE.
C			
C	USOL		
C		IMENSIONAL A	RRAY THAT
SPECIFIES THE			
C	VALUES (OF THE SOLUT	ION ALONG
THE BOUNDARIES.			
C	AT THE I	BOUNDARIES,	USOL IS
DEFINED BY		•	
C			
C	MBDCND	USOL(1,J)	
USOL(M+1,J)		. ,	
C			
С	0	*	*
С	1	U(A,YJ)	U(B,YJ)
С	2	U(A,YJ)	*

J=1,2,,N+1			
C	3	*	*
C	4	*	U(B,YJ)
C			
C	NBDCND	USOL(I,1)	
USOL(I,N+1)			
C			
C	0	*	*
C	1	U(XI,C)	U(XI,D)
С	2	U(XI,C)	*
I=1,2,,M+1			
С	3	*	*
C	4	*	U(XI,D)
C			
C	WHERE *	MEANS THE Q	UANTITIES
ARE NOT USED			
C	IN THE	SOLUTION.	
C			
C	IF IORD	ER=2, THE US	ER MAY
EQUIVALENCE GRHS		·	
C	AND USO	L TO SAVE SP	ACE. NOTE
THAT IN THIS			
C	CASE TH	E TABLES SPE	CIFYING
THE BOUNDARIES			
C	OF THE	GRHS AND USO	L ARRAYS
DETERMINE THE			
C	BOUNDAR:	IES UNIQUELY	EXCEPT AT
THE CORNERS.		~ ~	_
C	IF THE	TABLES CALL	FOR BOTH
G(X,Y) AND			
C	U(X,Y)	AT A CORNER '	THEN THE
SOLUTION MUST	O (11) 1	0011.	
C	BE CHOSI	EN. FOR EXA	MPTE, TF
MBDCND=2 AND	22 011001		· · · · · · · · · · · · · · · · · · ·
C	NBDCND=	4, THEN U(A,	C) -
U(A,D), U(B,D) MUST		1, 111111 O (A)	~ / /
C (A, D), 0 (B, D) F1031	BE CHOS	EN AT THE CO	RNERS IN
ADDITION			TIV TIV
C	TO G(B,	\sim)	
C	10 0 (D)	· , •	
C	TE TODD	ER=4, THEN T	HE TWO
ARRAYS, USOL AND	TI TONDI	⊔.,— 111€11 1.	1111 1 VVO
	CDUC M	ווכיי פני הדפיידי	мет
C	GKND, M	UST BE DISTI	INCI.

C USOL SHOULD BE DIMENSIONED IDMN BY AT LEAST C N+1 IN THE CALLING ROUTINE. C IDMN C IDMN C THE ROW (OR FIRST) DIMENSION OF THE ARRAYS C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C G C M C A fortran 90 derived TYPE (fishworkspace) variable C USE fish C C CUSE THAN CONTROL TO M+1. C C C C C CONTROL TO M+1. C C C C C C C C C C C C C C C C C C C		
IDMN BY AT LEAST C	C	USOL SHOULD BE DIMENSTONED
C IDMN C THE ARRAYS C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C W C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C USE fish C USE fish C THE declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		
C IDMN C THE ROW (OR FIRST) DIMENSION OF THE ARRAYS C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C G C A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program C calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C the first statement makes the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type	C	N+1 IN THE CALLING ROUTINE.
THE ROW (OR FIRST) DIMENSION OF THE ARRAYS C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C G C A fortran 90 derived TYPE that must be declared by the user. The first c user program c c C C C C C C C C C C C C C C C C C		
OF THE ARRAYS C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C W C A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program C C LING SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		
C GRHS AND USOL AS IT APPEARS IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C W C (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program c c USE fish C USE fish C USE fish C The declarative statement C must also be included in the user program C must also be included in the user program C must also be included in the user program C must also be included in the user program C must also be included in the user program C must also be included in the user program C must also be included in the user program C the first statement makes the fishpack module C user program calling SEPELI. The second statement C declares a derived type		THE ROW (OR FIRST) DIMENSION
IN THE PROGRAM C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C W C A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program C C USE fish C C USE fish C C THE declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		CRHS AND USOT, AS IT APPEARS
C CALLING SEPELI. THIS PARAMETER IS USED C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program C calling SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		OIVID AND OBOLI AD II AIILANO
C TO SPECIFY THE VARIABLE DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C W C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		CALLING SEPELI. THIS
DIMENSION OF GRHS C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C W C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C the first statement makes the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type	PARAMETER IS USED	
C AND USOL. IDMN MUST BE AT LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type	C	TO SPECIFY THE VARIABLE
LEAST 7 AND C GREATER THAN OR EQUAL TO M+1. C C A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program c calling SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		
C GREATER THAN OR EQUAL TO M+1. C W C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		AND USOL. IDMN MUST BE AT
C W A fortran 90 derived TYPE (fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program c calling SEPELI must be: C USE fish C USE fish C TYPE (fishworkspace) :: W C must also be included in the user program c must also be included in the user program c fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type		CDEATED TUAN OD FOLIAL TO M±1
C W C A fortran 90 derived TYPE (fishworkspace) variable C that must be declared by the user. The first C declarative statement in the user program C calling SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		GREATER THAN OR EQUAL TO MIT.
(fishworkspace) variable c that must be declared by the user. The first c declarative statement in the user program c calling SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		W
that must be declared by the user. The first c declarative statement in the user program c calling SEPELI must be: C USE fish C USE fish C THE declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program c must also be included in the user program C the first statement makes the fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement declares a derived type	С	A fortran 90 derived TYPE
user. The first c declarative statement in the user program c calling SEPELI must be: C USE fish C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type	(fishworkspace) variable	
c declarative statement in the user program c calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C the first statement makes the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		that must be declared by the
c calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C must also be included in the user program C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		
c calling SEPELI must be: C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the user program C the first statement makes the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		declarative statement in the
C USE fish C The declarative statement C TYPE (fishworkspace) :: W C must also be included in the user program C must also be included in the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type		calling SEPELI must be:
The declarative statement TYPE (fishworkspace) :: TYPE (fishworkspace) :: must also be included in the user program the first statement makes the fishpack module the defined in the file "fish.f" available to the the the the the second statement the declares a derived type		carring caract mass so.
The declarative statement TYPE (fishworkspace) :: TYPE (fishworkspace) :: TYPE (fishworkspace) :: Type (fishworkspace) :: The user program The first statement makes the fishpack module C defined in the file "fish.f" The second statement C declares a derived type	С	USE fish
C C C TYPE (fishworkspace):: W C C C Must also be included in the user program C The first statement makes the fishpack module C C C C Must also be included in the user program C The first statement makes the defined in the file "fish.f" available to the C U S TYPE (fishworkspace):: M C C Must also be included in the user program calling SEPELI. The second statement C D C D C D C D C D C D C D C D C D C	С	
C TYPE (fishworkspace) :: W C C must also be included in the user program C The first statement makes the fishpack module C defined in the file "fish.f" available to the C user program calling SEPELI. The second statement C declares a derived type	С	The declarative statement
W c c c must also be included in the user program c fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type		TYPE (fighterkenses)
c must also be included in the user program c The first statement makes the fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type		iie (iishworkspace) ::
c must also be included in the user program c The first statement makes the fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type		
c The first statement makes the fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type		must also be included in the
fishpack module c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type	user program	
c defined in the file "fish.f" available to the c user program calling SEPELI. The second statement c declares a derived type	С	The first statement makes the
available to the c user program calling SEPELI. The second statement c declares a derived type	fishpack module	
c user program calling SEPELI. The second statement c declares a derived type		defined in the file "fish.f"
The second statement c declares a derived type		user program calling SEPELT
c declares a derived type		aser program carring ourbur.
		declares a derived type
	variable (defined in	

С	the module "fish.f") which is
used internally	one meadle field, which is
С	in SEPELI to dynamically
allocate real and complex	
C An armon floor	work space used in solution.
An error flag	(IERROR = 20) is set if the
required work space	(111111011 - 20) 13 300 11 6110
C	allocation fails (for example
if N,M are too large)	_
C	Real and complex values are
set in the components	
c call to SEPELI. These	of W on a initial (INTL=0)
C C	must be preserved on non-
initial calls (INTL=1)	The second of th
С	to SEPELI. This eliminates
redundant calculations	
C ****	and saves compute time.
c **** calling SEPELI should	IMPORTANT! The user program
C	include the statement:
С	
С	CALL FISHFIN(W)
C	
is generated by	after the final approximation
C C	SEPELI. The will deallocate
the real and complex	
С	work space of W. Failure to
include this statement	
C	could result in serious
memory leakage.	
C	
	SOL
С	CONTAINS THE APPROXIMATE
SOLUTION TO THE	
C	ELLIPTIC EQUATION.
C APPROXIMATION TO U(XI,YJ)	USOL(I,J) IS THE
C C	FOR I=1,2,M+1 AND
J=1,2,,N+1.	
C	THE APPROXIMATION HAS ERROR

C WITH IORDER=2	O(DLX**2+DLY**2) IF CALLED
C C	AND O(DLX**4+DLY**4) IF
CALLED WITH	TODDED—4
C	IORDER=4.
С	W
c (fishworkspace) variable	The derived type
C (IISHWOIKSPACE) VAIIADIE	contains real and complex
values that must not	la la la la la la la Competita la
C called again with	be destroyed if SEPELI is
C	INTL=1.
C	PERTRB
C	IF A COMBINATION OF PERIODIC
OR DERIVATIVE	DOLIN DATA GONDATIONS
C	BOUNDARY CONDITIONS (I.E., ALPHA=BETA=0 IF
MBDCND=3;	(,
C SPECIFIED	GAMA=XNU=0 IF NBDCND=3) IS
C	AND IF THE COEFFICIENTS OF
U(X,Y) IN THE	
C ARE ZERO	SEPARABLE ELLIPTIC EQUATION
С	(I.E., $CF(X) = 0$ FOR X GREATER
THAN OR EQUAL C	TO A AND LESS THAN OR EQUAL
TO B;	TO A AND DESS THAN ON EQUAD
C	FF(Y)=0 FOR Y GREATER THAN OR
EQUAL TO C	AND LESS THAN OR EQUAL TO D)
THEN A	
C PERTRB IS A	SOLUTION MAY NOT EXIST.
C C	CONSTANT CALCULATED AND
SUBTRACTED FROM	MILE DICUM HAND CIDE OF THE
C MATRIX EQUATIONS	THE RIGHT-HAND SIDE OF THE
С	GENERATED BY SEPELI WHICH
INSURES THAT A	SOLUTION EXISTS. SEPELI THEN
C	DOTITION EVIDIO. SELETT THEN

COMPUTES THIS	
C	SOLUTION WHICH IS A WEIGHTED
MINIMAL LEAST	
C	SQUARES SOLUTION TO THE
ORIGINAL PROBLEM.	~
C	
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS OR FAILURE TO FIND
A SOLUTION	TAINMETERS ON PAIDONE TO PIND
	- 0 NO EDDOD
C	= 0 NO ERROR
C	= 1 IF A GREATER THAN B OR C
GREATER THAN D	0
C	= 2 IF MBDCND LESS THAN 0 OR
MBDCND GREATER	
C	THAN 4
C	= 3 IF NBDCND LESS THAN 0 OR
NBDCND GREATER	
C	THAN 4
C	= 4 IF ATTEMPT TO FIND A
SOLUTION FAILS.	
C	(THE LINEAR SYSTEM
GENERATED IS NOT	
C	DIAGONALLY DOMINANT.)
C	= 5 IF IDMN IS TOO SMALL
C	(SEE DISCUSSION OF IDMN)
С	= 6 IF M IS TOO SMALL OR TOO
LARGE	
C	(SEE DISCUSSION OF M)
C	= 7 IF N IS TOO SMALL (SEE
DISCUSSION OF N)	, 11 11 10 100 811111 (811
C	= 8 IF IORDER IS NOT 2 OR 4
C	= 9 IF INTL IS NOT 0 OR 1
C	= 10 IF AFUN*DFUN LESS THAN
OR EQUAL TO 0	- 10 IF AFON DEON LESS INAN
C C	EOD COME THEEDTOD MEGI
	FOR SOME INTERIOR MESH
POINT (XI, YJ)	00 75
C	= 20 If the dynamic
allocation of real and	
C	complex work space in
the derived type	
C	(fishworkspace) variable
W fails (e.g.,	

C	if N,M are too large for
the platform used)	
C	
С	NOTE (CONCERNING IERROR=4):
FOR THE	
С	COEFFICIENTS INPUT THROUGH
COFY, COFY,	
C	THE DISCRETIZATION MAY LEAD
TO A BLOCK	
C	TRIDIAGONAL LINEAR SYSTEM
WHICH IS NOT	
C C	DIACONALLY DOMINANT /FOD
	DIAGONALLY DOMINANT (FOR
EXAMPLE, THIS	III DDENIG TE GEIRI O AND
C	HAPPENS IF CFUN=0 AND
BFUN/(2.*DLX) GREATER	
C	THAN AFUN/DLX**2). IN THIS
CASE SOLUTION	
C	MAY FAIL. THIS CANNOT HAPPEN
IN THE LIMIT	
C	AS DLX, DLY APPROACH ZERO.
HENCE, THE	
C	CONDITION MAY BE REMEDIED BY
TAKING LARGER	
C	VALUES FOR M OR N.
C	
C SPECIAL CONDITIONS	SEE COFX, COFY ARGUMENT
DESCRIPTIONS ABOVE.	,
C	
C I/O	NONE
C 1/0	110111
C PRECISION	CINCIE
	SINGLE
C	
C	
C REQUIRED FILES	blktri.f,comf.f,sepaux.f,fish.f
C	
C LANGUAGE	Fortran 90
C	
C HISTORY	DEVELOPED AT NCAR DURING 1975-
76 BY	
C	JOHN C. ADAMS OF THE SCIENTIFIC
COMPUTING	
C	DIVISION. RELEASED ON NCAR'S
PUBLIC SOFTWARE	
С	LIBRARIES IN JANUARY 1980.

Revised in June	
C	2004 using Fortan 90
dynamically allocated wo	
C	space and derived data types to
eliminate mixed	mode conflicts in the conlice
c versions. All	mode conflicts in the earlier
C C	statement labels, arithmetic if
statements and	scatement labers, affilmetic if
C	computed GO TO statements have
been removed from	compaced to 10 seatements nave
C	the current version of SEPELI.
C	
C PORTABILITY	FORTRAN 90
C	
C ALGORITHM	SEPELI AUTOMATICALLY
DISCRETIZES THE	
С	SEPARABLE ELLIPTIC EQUATION
WHICH IS THEN	
C	SOLVED BY A GENERALIZED CYCLIC
REDUCTION	
C	ALGORITHM IN THE SUBROUTINE,
BLKTRI. THE	EQUIDMU ODDED COLUMNOM TO
C OBTAINED USING	FOURTH-ORDER SOLUTION IS
C C OBTAINED USING	'DEFERRED CORRECTIONS' WHICH IS
DESCRIBED	DEFEIGED CONTECTIONS WITCH 15
C	AND REFERENCED IN SECTIONS,
REFERENCES AND	
C	METHOD.
C	
C TIMING	THE OPERATIONAL COUNT IS
PROPORTIONAL TO	
С	M*N*LOG2(N).
C	
C ACCURACY	THE FOLLOWING ACCURACY RESULTS
WERE OBTAINED	
C	using 64 bit floating point
arithmetic. Note	
C	THAT THE FOURTH-ORDER accuracy
is not realized	INITI THE MECH TO ALEE ALAST
C refined.	UNTIL THE MESH IS sufficiently
C C	

С			SECOND-ORDER
FOURTH-ORDER			
С	M	N	ERROR
ERROR			
С			
С	6	6	6.8E-1
1.2E0			
С	14	14	1.4E-1
1.8E-1			
С	30	30	3.2E-2
9.7E-3			
С	62	62	7.5E-3
3.0E-4			
С	126	126	1.8E-3
3.5E-6			
С			
С			
C REFERENCES	KELLER,	н.В.,	NUMERICAL METHODS
FOR TWO-POINT			
C	BOUNDAR	Y-VALU	JE PROBLEMS,
BLAISDEL (1968),			
C	WALTHAM	, MASS	· .
C			
С	SWARZTR	AUBER,	P., AND R. SWEET
(1975):			
C	EFFICIE	NT FOF	RTRAN SUBPROGRAMS
FOR THE			
C	SOLUTIO	N OF E	ELLIPTIC PARTIAL
DIFFERENTIAL			
С			ICAR TECHNICAL NOTE
C			9, PP. 135-137.
C******	*****	*****	******

SEPX4

```
C file sepx4.txt (documentation for the FISHPACK
```

```
solver SEPX4)
C
                        copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
                                     of
```

```
С
C
                the National Center for Atmospheric
Research
C
                        Boulder, Colorado (80307)
U.S.A.
С
С
                           which is sponsored by
С
С
                     the National Science Foundation
      *
С
      SUBROUTINE SEPX4
(IORDER, A, B, M, MBDCND, BDA, ALPHA, BDB, BETA, C, D, N,
NBDCND, BDC, BDD, COFX, GRHS, USOL, IDMN, PERTRB,
С
                         IERROR)
С
С
С
C DIMENSION OF
                         BDA(N+1), BDB(N+1), BDC(M+1),
BDD(M+1),
C ARGUMENTS
                         USOL(IDMN,N+1),
GRHS (IDMN, N+1),
C
C
C LATEST REVISION June 2004
C PURPOSE
                          SEPX4 SOLVES FOR EITHER THE
SECOND-ORDER
                          FINITE DIFFERENCE APPROXIMATION
C
OR A
С
                          FOURTH-ORDER APPROXIMATION TO A
SEPARABLE
                          ELLIPTIC EQUATION
```

```
С
AF(X) *UXX+BF(X) *UX+CF(X) *U+UYY = G(X,Y)
C
С
                          ON A RECTANGLE (X GREATER THAN
OR EQUAL TO
                          A AND LESS THAN OR EQUAL TO B,
Y GREATER THAN
                         OR EQUAL TO C AND LESS THAN OR
EQUAL TO D).
                          ANY COMBINATION OF PERIODIC OR
MIXED BOUNDARY
                          CONDITIONS IS ALLOWED. IF
BOUNDARY
                          CONDITIONS IN THE X DIRECTION
ARE PERIODIC
                          (SEE MBDCND=0 BELOW) THEN THE
COEFFICIENTS
                          MUST SATISFY
С
С
                            AF(X) = C1, BF(X) = 0, CF(X) = C2 FOR
ALL X.
C
                         HERE C1, C2 ARE CONSTANTS,
C1.GT.0.
С
                          THE POSSIBLE BOUNDARY
CONDITIONS ARE:
                          IN THE X-DIRECTION:
                            (0) PERIODIC, U(X+B-
A, Y) = U(X, Y) FOR
                                ALL Y, X
С
                            (1) U(A,Y), U(B,Y) ARE
SPECIFIED FOR ALL Y
                            (2) U(A, Y),
DU(B,Y)/DX+BETA*U(B,Y) ARE
С
                                SPECIFIED FOR ALL Y
                            (3)
C
DU(A,Y)/DX+ALPHA*U(A,Y),DU(B,Y)/DX+
                                BETA*U(B,Y) ARE SPECIFIED
FOR ALL Y
                            (4)
C
DU(A,Y)/DX+ALPHA*U(A,Y),U(B,Y) ARE
                              SPECIFIED FOR ALL Y
```

```
C
С
                          IN THE Y-DIRECTION:
C
                             (0) PERIODIC, U(X,Y+D-
C) = U(X,Y) FOR ALL X,Y
                             (1) U(X,C),U(X,D) ARE
SPECIFIED FOR ALL X
                             (2) U(X,C), DU(X,D)/DY ARE
SPECIFIED FOR
С
                                 ALL X
                             (3) DU(X,C)/DY, DU(X,D)/DY ARE
SPECIFIED FOR
                                ALL X
C
                            (4) DU(X,C)/DY,U(X,D) ARE
SPECIFIED FOR
                                ALL X
С
C USAGE
                          CALL
SEPX4 (IORDER, A, B, M, MBDCND, BDA, ALPHA, BDB,
BETA, C, D, N, NBDCND, BDC, BDD, COFX,
GRHS, USOL, IDMN, W, PERTRB, IERROR)
C
C ARGUMENTS
C ON INPUT
                          IORDER
C
                            = 2 IF A SECOND-ORDER
APPROXIMATION IS
C
                                 SOUGHT
С
                            = 4 IF A FOURTH-ORDER
APPROXIMATION IS
C
                                 SOUGHT
C
c *** caution ***
                            GRHS SHOULD BE RESET IF SEPX4
WAS FIRST CALLED
                            WITH IORDER=2 AND WILL BE
CALLED AGAIN WITH
                            IORDER=4. VALUES IN GRHS ARE
DESTROYED BY THE
                            IORDER=2 CALL.
C
С
С
                          A,B
C
                            THE RANGE OF THE X-
INDEPENDENT VARIABLE,
```

C ROLLAT MO A	I.E., X IS GREATER THAN OR
EQUAL TO A	AND LESS THAN OR EQUAL TO B.
A MUST BE	THE PLOS THE OR EXCILE TO B.
С	LESS THAN B.
С	
С	M
C	THE NUMBER OF PANELS INTO
WHICH THE	
C HENCE,	INTERVAL (A, B) IS SUBDIVIDED.
C	THERE WILL BE M+1 GRID POINTS
IN THE X-	THERE WILL BE THE GREET TOTAL
С	DIRECTION GIVEN BY XI=A+(I-
1) *DLX	
C	FOR $I=1,2,\ldots,M+1$ WHERE
DLX=(B-A)/M IS	
C TECC BUAN	THE PANEL WIDTH. M MUST BE
LESS THAN C	IDMN AND GREATER THAN 5.
C	IDM AND GREATER HAN J.
C	MBDCND
С	INDICATES THE TYPE OF
BOUNDARY CONDITION	
С	AT X=A AND X=B
C	= 0 IF THE SOLUTION IS
PERIODIC IN X, I.E.,	
C ALL Y,X	U(X+B-A,Y)=U(X,Y) FOR
C C	= 1 IF THE SOLUTION IS
SPECIFIED AT X=A	
С	AND $X=B$, I.E., $U(A,Y)$ AND
U(B,Y) ARE	
С	SPECIFIED FOR ALL Y
C	= 2 IF THE SOLUTION IS
SPECIFIED AT X=A	AND THE DOLLARY
C CONDITION IS MIXED AT	AND THE BOUNDARY
CONDITION IS MIXED AT	X=B, I.E., U(A,Y) AND
C	DU(B,Y)/DX+BETA*U(B,Y)
ARE SPECIFIED	
С	FOR ALL Y
С	= 3 IF THE BOUNDARY
CONDITIONS AT X=A AND	

C	X=B ARE MIXED, I.E.,
C	DU(A,Y)/DX+ALPHA*U(A,Y)
AND	
C	DU(B,Y)/DX+BETA*U(B,Y)
ARE SPECIFIED	
C	FOR ALL Y
C	= 4 IF THE BOUNDARY CONDITION
AT X=A IS	
C	MIXED AND THE SOLUTION IS
SPECIFIED	
C	AT X=B, I.E.,
DU(A,Y)/DX+ALPHA*U(A,Y)	
C	AND U(B,Y) ARE SPECIFIED
FOR ALL Y	
C	
C	BDA
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1 THAT	
C	SPECIFIES THE VALUES OF
C	DU(A,Y)/DX+ALPHA*U(A,Y)AT
X=A, WHEN	
C	MBDCND=3 OR 4.
C	BDA(J) =
DU(A,YJ)/DX+ALPHA*U(A,YJ)	,
C	J=1,2,,N+1
C	WHEN MBDCND HAS ANY OTHER
VALUE, BDA IS	
C	A DUMMY PARAMETER.
C	
С	ALPHA
С	THE SCALAR MULTIPLYING THE
SOLUTION IN CASE	
С	OF A MIXED BOUNDARY CONDITION
AT X=A	
C	(SEE ARGUMENT BDA). IF
MBDCND IS NOT EQUAL	
C	TO EITHER 3 OR 4, THEN ALPHA
IS A DUMMY	
С	PARAMETER.
C	
C	BDB
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH N+1 THAT	
C	SPECIFIES THE VALUES OF

```
C
                           DU(B,Y)/DX + BETA*U(B,Y) AT
X=B.
С
                           WHEN MBDCND=2 OR 3
C
                           BDB(J) =
DU(B,YJ)/DX+BETA*U(B,YJ),
                           J=1,2,...,N+1
                           WHEN MBDCND HAS ANY OTHER
VALUE, BDB IS
С
                           A DUMMY PARAMETER.
С
                         BETA
С
С
                           THE SCALAR MULTIPLYING THE
SOLUTION IN
                          CASE OF A MIXED BOUNDARY
CONDITION AT X=B
                           (SEE ARGUMENT BDB). IF
MBDCND IS NOT EQUAL
                           TO 2 OR 3, THEN BETA IS A
DUMMY PARAMETER.
С
                         C,D
                           THE RANGE OF THE Y-
INDEPENDENT VARIABLE,
                           I.E., Y IS GREATER THAN OR
EQUAL TO C AND
                           LESS THAN OR EQUAL TO D. C
MUST BE LESS
                           THAN D.
C
С
С
                         N
                           THE NUMBER OF PANELS INTO
WHICH THE
                           INTERVAL (C,D) IS SUBDIVIDED.
C
HENCE,
                           THERE WILL BE N+1 GRID POINTS
IN THE Y-
                           DIRECTION GIVEN BY YJ=C+(J-
1) *DLY FOR
                           J=1,2,...,N+1 WHERE DLY=(D-
C)/N IS THE
                           PANEL WIDTH. IN ADDITION, N
MUST BE
C
                           GREATER THAN 4.
```

C	NBDCND
C	INDICATES THE TYPES OF
BOUNDARY CONDITIONS	
C	AT Y=C AND Y=D
C	= 0 IF THE SOLUTION IS
	O II IIII BODOITON IB
PERIODIC IN Y,	
C	I.E., $U(X,Y+D-C)=U(X,Y)$
FOR ALL X, Y	
C	= 1 IF THE SOLUTION IS
SPECIFIED AT Y=C	
C	AND $Y = D$, I.E., $U(X,C)$
	AND $I = D_i \cup I \cup $
AND U(X, D)	
C	ARE SPECIFIED FOR ALL X
C	= 2 IF THE SOLUTION IS
SPECIFIED AT Y=C	
C	AND THE BOUNDARY
CONDITION IS MIXED	
C	AT $Y=D$, I.E., $DU(X,C)/DY$
AND U(X, D)	
C	ARE SPECIFIED FOR ALL X
C	= 3 IF THE BOUNDARY
CONDITIONS ARE MIXED	
C	AT Y=CAND Y=D I.E.,
	•
C	DU(X,D)/DY AND $DU(X,D)/DY$
ARE	
C	SPECIFIED FOR ALL X
C	= 4 IF THE BOUNDARY CONDITION
IS MIXED	
C	AT Y=C AND THE SOLUTION
	AT T-C AND THE SOLUTION
IS SPECIFIED	
C	AT Y=D, I.E.
DU(X,C)/DY+GAMA*U(X,C)	
C	AND U(X,D) ARE SPECIFIED
FOR ALL X	, ,
C	
	DDG
C	BDC
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT	
C	SPECIFIES THE VALUE
DU(X,C)/DY AT Y=C.	
C	THEN ADDONE O OF A
C	WHEN NBDCND=3 OR 4
C	BDC(I) = DU(XI,C)/DY
I=1,2,,M+1.	

C	WHEN NBDCND HAS ANY OTHER
VALUE, BDC IS	
C	A DUMMY PARAMETER.
C	
С	BDD
C	A ONE-DIMENSIONAL ARRAY OF
LENGTH M+1 THAT	
C DII (Y D) (DY AM Y-D	SPECIFIED THE VALUE OF
DU(X,D)/DY AT Y=D.	
C	WHEN NBDCND=2 OR 3
C	BDD(I)=DU(XI,D)/DY
I=1,2,,M+1.	(_, (, , ,)
C	
C	WHEN NBDCND HAS ANY OTHER
VALUE, BDD IS	
C	A DUMMY PARAMETER.
C	COTIL
C	COFX A USER-SUPPLIED SUBPROGRAM
WITH PARAMETERS	A USER-SUPPLIED SUBPROGRAM
C C	X, AFUN, BFUN, CFUN WHICH
RETURNS THE	n, mon, bron, or on milon
С	VALUES OF THE X-DEPENDENT
COEFFICIENTS	
C	AF(X), $BF(X)$, $CF(X)$ IN THE
ELLIPTIC	
C	EQUATION AT X. IF BOUNDARY
CONDITIONS IN	MILE V DIDECHION ARE DEDICATE
C THEN THE	THE X DIRECTION ARE PERIODIC
C	COEFFICIENTS MUST SATISFY
AF(X) = C1, BF(X) = 0,	
C	CF(X) = C2 FOR ALL X. HERE
C1.GT.0	
С	AND C2 ARE CONSTANTS.
С	
C	NOTE THAT COFX MUST BE
DECLARED EXTERNAL	THE MILE CALL THE DOLLMAND
C	IN THE CALLING ROUTINE.
C	GRHS
C	A TWO-DIMENSIONAL ARRAY THAT
-	

SPECIFIES THE C	VALUES (OF THE RIGHT	-HAND SIDE
OF THE C I.E., GRHS (I, J) = G(XI, YI), C THE	ELLIPTIC EQUATION,		
	FOR I=2,,M, J=2,,N. AT		
C BY C	BOUNDAR:	IES, GRHS IS	DEFINED
C GRHS (M+1,J)	MBDCND	GRHS (1, J)	
C 			
C C	0 1	G(A,YJ) *	G(B,YJ) *
C J=1,2,,N+1	2	*	G(B,YJ)
C C	3 4	G(A,YJ) G(A,YJ)	G(B,YJ) *
C C GRHS(I,N+1)	NBDCND	GRHS(I,1)	
C			
C C	0 1	G(XI,C) *	G(XI,D) *
C I=1,2,,M+1	2	*	G(XI,D)
C C	3 4	G(XI,C) G(XI,C)	G(XI,D) *
C C	WHERE *	MEANS THESE	QUANTITES
ARE NOT USED. C IDMN BY AT LEAST	GRHS SHO	OULD BE DIME	NSIONED
C C	N+1 IN S	THE CALLING	ROUTINE.
c *** caution WAS FIRST CALLED	GRHS SHO	OULD BE RESE	T IF SEPX4
C CALLED AGAIN WITH	WITH IO	RDER=2 AND W	TLL BE
C DESTROYED BY THE	IORDER=	4. VALUES I	N GRHS ARE

C C C USOL C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE C VALUES OF THE SOLUTION ALON THE BOUNDARIES. C AT THE BOUNDARIES, USOL IS	
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE C VALUES OF THE SOLUTION ALON THE BOUNDARIES. C AT THE BOUNDARIES, USOL IS	
SPECIFIES THE C VALUES OF THE SOLUTION ALON THE BOUNDARIES. C AT THE BOUNDARIES, USOL IS	
THE BOUNDARIES. C AT THE BOUNDARIES, USOL IS	IG
C AT THE BOUNDARIES, USOL IS	
,	
Defetaled DV	
DEFINED BY C	
C MBDCND USOL(1,J)	
USOL(M+1,J)	
C	
C 0 * *	
C 1 U(A,YJ) U(B,YJ	-)
C 2 U(A, YJ) *	,
J=1,2,,N+1	
C 3 * * *	-\
C 4 * U(B, Y))
C NBDCND USOL(I,1)	
USOL(I,N+1)	
C	
 C	
C 1 U(XI,C) U(XI,I))
C 2 U(XI,C) *	,
I=1,2,,M+1	
C 3 * * II(XT_I	
C 4 * U(XI,I	')
C WHERE * MEANS THE QUANTITES	
ARE NOT USED	
C IN THE SOLUTION.	
C IF IORDER=2, THE USER MAY	
C IF IORDER=2, THE USER MAY EQUIVALENCE GRHS	
C AND USOL TO SAVE SPACE. NO	TE
THAT IN THIS	
C CASE THE TABLES SPECIFYING	
THE BOUNDARIES C OF THE GRHS AND USOL ARRAYS	
DETERMINE THE	

C THE CODNEDC	BOUNDARIES UNIQUELY EXCEPT AT
THE CORNERS.	IF THE TABLES CALL FOR BOTH
G(X,Y) AND	
C	U(X,Y) AT A CORNER THEN THE
SOLUTION MUST	
C	BE CHOSEN.
C	FOR EXAMPLE, IF MBDCND=2 AND
NBDCND=4,	
C	THEN $U(A,C)$, $U(A,D)$, $U(B,D)$
MUST BE CHOSEN	
C	AT THE CORNERS IN ADDITION TO
G(B,C).	
C	
C	IF IORDER=4, THEN THE TWO
ARRAYS, USOL AND	
C	GRHS, MUST BE DISTINCT.
C	
C	USOL SHOULD BE DIMENSIONED
IDMN BY AT LEAST	
C	N+1 IN THE CALLING ROUTINE.
C	
C	IDMN
C	IDMN THE ROW (OR FIRST) DIMENSION
C OF THE ARRAYS C	
C OF THE ARRAYS	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS
C OF THE ARRAYS C IN THE PROGRAM C	THE ROW (OR FIRST) DIMENSION
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1.
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C C C C C C ON OUTPUT	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C C C ON OUTPUT C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1.
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C SOLUTION TO THE	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL CONTAINS THE APPROXIMATE
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C SOLUTION TO THE C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C C C C C C C C SOLUTION TO THE C IS THE	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL CONTAINS THE APPROXIMATE ELLIPTIC EQUATION. USOL(I,J)
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C C C C SOLUTION TO THE C IS THE C	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL CONTAINS THE APPROXIMATE
C OF THE ARRAYS C IN THE PROGRAM C PARAMETER IS USED C DIMENSION OF GRHS C LEAST 7 AND C C C C C C C C C C C C C SOLUTION TO THE C IS THE	THE ROW (OR FIRST) DIMENSION GRHS AND USOL AS IT APPEARS CALLING SEPELI. THIS TO SPECIFY THE VARIABLE AND USOL. IDMN MUST BE AT GREATER THAN OR EQUAL TO M+1. USOL CONTAINS THE APPROXIMATE ELLIPTIC EQUATION. USOL(I,J)

APPROXIMATION HAS	
C	ERROR O(DLX**2+DLY**2) IF
CALLED WITH	
C	IORDER=2 AND O(DLX**4+DLY**4)
IF CALLED	
C	WITH IORDER=4.
C	
C	PERTRB
C	IF A COMBINATION OF PERIODIC
OR DERIVATIVE	
C	BOUNDARY CONDITIONS (I.E.,
ALPHA=BETA=0 IF	
C	MBDCND=3) IS SPECIFIED AND IF
CF(X) = 0 FOR	
C	ALL X THEN A SOLUTION TO THE
DISCRETIZED	
C	MATRIX EQUATION MAY NOT EXIST
C	(REFLECTING THE NON-
UNIQUENESS OF SOLUTIONS	mo mun ppn)
C	TO THE PDE).
C CALCULATED AND	PERTRB IS A CONSTANT
CALCULATED AND	CUDEDACED EDOM BUE DICUE
HAND SIDE OF THE	SUBTRACTED FROM THE RIGHT
	MADDLY EQUADION INCUDING DUE
C EXISTENCE OF A	MATRIX EQUATION INSURING THE
C C	SOLUTION. SEPX4 COMPUTES
THIS SOLUTION	SOLUTION. SEPA4 COMPUTES
C C	WHICH IS A WEIGHTED MINIMAL
LEAST SQUARES	WITCH 15 A WEIGHTED MINIME
C C	SOLUTION TO THE ORIGINAL
PROBLEM. IF	SOLUTION TO THE ONIOTIVAL
C	SINGULARITY IS NOT DETECTED
PERTRB=0.0 IS	
C	RETURNED BY SEPX4.
C	
C	IERROR
C	AN ERROR FLAG THAT INDICATES
INVALID INPUT	
C	PARAMETERS OR FAILURE TO FIND
A SOLUTION	2 22. 22.20.
C	
C	= 0 NO ERROR
C	= 1 IF A GREATER THAN B OR C

CDEAMED		
GREATER		milyn D
C	0	THAN D
C	= 2	IF MBDCND LESS THAN 0 OR
MBDCND		
C	_	GREATER THAN 4
C	= 3	IF NBDCND LESS THAN 0 OR
NBDCND		
C		GREATER THAN 4
C	= 4	IF ATTEMPT TO FIND A
SOLUTION FAILS.		
C		(THE LINEAR SYSTEM
GENERATED IS NOT		
С		DIAGONALLY DOMINANT.)
C	= 5	IF IDMN IS TOO SMALL
(SEE DISCUSSION		
С		OF IDMN)
С	= 6	IF M IS TOO SMALL OR TOO
LARGE		
С		(SEE DISCUSSION OF M)
C	= 7	IF N IS TOO SMALL (SEE
DISCUSSION OF N)		
C	= 8	IF IORDER IS NOT 2 OR 4
C		IF INTL IS NOT 0 OR 1
C		IF AFUN IS LESS THAN OR
EQUAL TO ZERO	10	II III OIV IO LLOO III IV OIV
C ZEIKO		FOR SOME INTERIOR MESH
POINT XI SOME		
C		INTERIOR MESH POINT
(XI,YJ)		INIBICION PESTI TOTAL
(A1,10)	- 12	IF MBDCND=0 AND
AF(X) = CF(X) = CONSTANT	- 12	IF PIDDCIND-U AND
C		OR BF(X)=0 FOR ALL X IS
NOT TRUE.		OR BF (A) =0 FOR ALL A 15
	- 20	If the dimension
C	- 20	If the dynamic
allocation of real and		
C		complex work space
required for solution		5-11- (5 1- 1- 1- 1- 1- 1- 1- 1- 1- 1-
C		fails (for example if
N,M are too large		
C		for your computer)
C		
C SPECIAL CONDITIONS	NONE	
C		
C I/O	NONE	

```
C REQUIRED files
fish.f,comf.f,genbun.f,gnbnaux.f,sepaux.f
С
C PRECISION
                        SINGLE
C
C LANGUAGE
                FORTRAN 90
C HISTORY
                    SEPX4 WAS DEVELOPED AT NCAR BY
JOHN C.
                       ADAMS OF THE SCIENTIFIC
COMPUTING DIVISION
                        IN OCTOBER 1978. THE BASIS OF
THIS CODE IS
                       NCAR ROUTINE SEPELI. BOTH
PACKAGES WERE
                        RELEASED ON NCAR'S PUBLIC
LIBRARIES IN
                        JANUARY 1980. SEPX4 was
modified in June 2004
                        incorporating fortran 90
dynamical storage
                        allocation for work space
requirements
C PORTABILITY
                       FORTRAN 90
C ALGORITHM
                       SEPX4 AUTOMATICALLY DISCRETIZES
THE SEPARABLE
                        ELLIPTIC EQUATION WHICH IS THEN
SOLVED BY A
                        GENERALIZED CYCLIC REDUCTION
ALGORITHM IN THE
                        SUBROUTINE POIS. THE FOURTH
ORDER SOLUTION
                        IS OBTAINED USING THE TECHNIQUE
OF DEFFERRED
                        CORRECTIONS REFERENCED BELOW.
С
C TIMING
                        WHEN POSSIBLE, SEPX4 SHOULD BE
USED INSTEAD
                        OF PACKAGE SEPELI. THE
```

```
INCREASE IN SPEED
                      IS AT LEAST A FACTOR OF THREE.
C
C REFERENCES
                      KELLER, H.B., NUMERICAL METHODS
FOR TWO-POINT
                      BOUNDARY-VALUE PROBLEMS,
BLAISDEL (1968),
                      WALTHAM, MASS.
С
                      SWARZTRAUBER, P., AND R. SWEET
(1975):
                      EFFICIENT FORTRAN SUBPROGRAMS
FOR THE
                      SOLUTION OF ELLIPTIC PARTIAL
DIFFERENTIAL
                      EQUATIONS. NCAR TECHNICAL NOTE
                       NCAR-TN/IA-109, PP. 135-137.
C**************
******
```

TBLKTRI

```
С
C
     file tblktri.f
С
C
* * * * * * *
С
                        copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
C
*
С
                            all rights reserved
```

```
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
*
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
С
*
                     the National Science Foundation
```

```
С
С
С
      PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE BLKTRI
TO
С
      SOLVE THE EQUATION
С
С
      .5/S*(D/DS)(.5/S*DU/DS)+.5/T*(D/DT)(.5/T*DU/DT)
C
(1)
С
                     = 15/4*S*T*(S**4+T**4)
С
      ON THE RECTANGLE 0 .LT. S .LT. 1 AND 0 .LT. T .LT.
С
1
С
      WITH THE BOUNDARY CONDITIONS
С
С
      U(0,T) = 0
С
                              0 .LE. T .LE. 1
С
      U(1,T) = T**5
С
С
      AND
С
С
      U(S,0) = 0
С
                             0 .LE. S .LE. 1
С
      U(S,1) = S**5
С
С
      THE EXACT SOLUTION OF THIS PROBLEM IS U(S,T) =
(S*T)**5
C
      DEFINE THE INTEGERS M = 50 AND N = 63. THEN DEFINE
С
THE
С
      GRID INCREMENTS DELTAS = 1/(M+1) AND DELTAT =
1/(N+1).
С
С
      THE GRID IS THEN GIVEN BY S(I) = I*DELTAS FOR I = I*DELTAS
1, ..., M
      AND T(J) = J*DELTAT FOR J = 1, ..., N.
C
С
С
      THE APPROXIMATE SOLUTION IS GIVEN AS THE SOLUTION
TO
      THE FOLLOWING FINITE DIFFERENCE APPROXIMATION OF
```

```
EOUATION (1).
С
      .5/(S(I)*DELTAS)*((U(I+1,J)-
U(I,J))/(2*S(I+.5)*DELTAS)
                     -(U(I,J)-U(I-1,J))/(2*S(I-
.5) *DELTAS))
     +.5/(T(I)*DELTAT)*((U(I,J+1)-
U(I,J))/(2*T(I+.5)*DELTAT) (2)
C
                     -(U(I,J)-U(I,J-1))/(2*T(I-
.5) *DELTAT))
               = 15/4*S(I)*T(J)*(S(I)**4+T(J)**4)
C
С
С
             WHERE S(I+.5) = .5*(S(I+1)+S(I))
С
                   S(I-.5) = .5*(S(I)+S(I-1))
С
                   T(I+.5) = .5*(T(I+1)+T(I))
С
                   T(I-.5) = .5*(T(I)+T(I-1))
С
С
     THE APPROACH IS TO WRITE EQUATION (2) IN THE FORM
С
С
     AM(I) *U(I-1, J) +BM(I) *U(I, J) +CM(I) *U(I+1, J)
С
       +AN(J)*U(I,J-1)+BN(J)*U(I,J)+CN(J)*U(I,J+1)
(3)
C
           = Y(I,J)
С
C
     AND THEN CALL SUBROUTINE BLKTRI TO DETERMINE
U(I,J)
C
С
С
     PROGRAM TBLKTRI
     USE fish
     implicit none
     TYPE (fishworkspace) :: w
C-----
   Local Variables
C-----
     INTEGER :: IFLG, NP, N, MP, M, IDIMY, I, J, IERROR
     REAL , DIMENSION (75, 105) :: Y
     REAL , DIMENSION (75) :: AM, BM, CM
     REAL , DIMENSION (105) :: AN, BN, CN
     REAL , DIMENSION (75) :: S
     REAL , DIMENSION (105) :: T
REAL::DELTAS, DELTAT, HDS, TDS, TEMP1, TEMP2, TEMP3, HDT, TDT, ER
```

```
R,Z
      IFLG = 0
      NP = 1
      N = 63
      MP = 1
     M = 50
      IDIMY = 75
C
С
      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING THE
С
      COEFFICIENTS AND THE ARRAY Y.
C
      DELTAS = 1./FLOAT(M + 1)
      DO I = 1, M
         S(I) = FLOAT(I)*DELTAS
      END DO
      DELTAT = 1./FLOAT(N + 1)
      DO J = 1, N
         T(J) = FLOAT(J)*DELTAT
      END DO
C
C
      COMPUTE THE COEFFICIENTS AM, BM, CM CORRESPONDING TO
THE S DIRECTION
      HDS = DELTAS/2.
      TDS = DELTAS + DELTAS
      DO 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))
      END DO
С
      COMPUTE THE COEFFICIENTS AN, BN, CN CORRESPONDING TO
THE T DIRECTION
С
      HDT = DELTAT/2.
      TDT = DELTAT + DELTAT
      DO 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))
      END DO
C
С
      COMPUTE RIGHT SIDE OF EQUATION
С
     DO J = 1, N
         Y(:M,J) = 3.75*S(:M)*T(J)*(S(:M)**4+T(J)**4)
      END DO
С
C
      THE NONZERO BOUNDARY CONDITIONS ENTER THE LINEAR
SYSTEM VIA
      THE RIGHT SIDE Y(I,J). IF THE EQUATIONS (3) GIVEN
ABOVE
     ARE EVALUATED AT I=M AND J=1,...,N THEN THE TERM
CM(M) *U(M+1,J)
      IS KNOWN FROM THE BOUNDARY CONDITION TO BE
CM(M) *T(J) **5.
      THEREFORE THIS TERM CAN BE INCLUDED IN THE RIGHT
SIDE Y(M,J).
      THE SAME ANALYSIS APPLIES AT J=N AND I=1,..,M.
NOTE THAT THE
      CORNER AT J=N, I=M INCLUDES CONTRIBUTIONS FROM BOTH
BOUNDARIES.
      Y(M, :N) = Y(M, :N) - CM(M) *T(:N) **5
      Y(:M,N) = Y(:M,N) - CN(N)*S(:M)**5
С
С
      DETERMINE THE APPROXIMATE SOLUTION U(I, J)
C
      CALL
BLKTRI (IFLG, NP, N, AN, BN, CN, MP, M, AM, BM, CM, IDIMY, Y, IERROR, W
      IFLG = IFLG + 1
      DO WHILE (IFLG -1 \le 0)
         CALL BLKTRI (IFLG, NP, N, AN, BN, CN, MP, M,
AM, BM, CM, IDIMY
          , Y, IERROR, W)
         IFLG = IFLG + 1
      END DO
      ERR = 0.
     DO J = 1, N
```

```
DO I = 1, M
           Z = ABS(Y(I,J) - (S(I)*T(J))**5)
           ERR = AMAX1(Z, ERR)
        END DO
     END DO
     Print earlier output from platforms with 32 and 64
bit floating point
     arithemtic followed by the output from this
computer
     WRITE (*, *) ' BLKTRI TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.6478E-05'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.2737E-02'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
     release dynamically allocated work space
     CALL FISHFIN (W)
     STOP
     END PROGRAM TBLKTRI
```

TCBLKTRI

```
copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
С
Research
                             all rights reserved
С
                            FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
```

```
C
                    Boulder, Colorado (80307)
U.S.A.
C
С
                        which is sponsored by
С
С
                   the National Science Foundation
С
С
     PROGRAM TCBLKTRI
     USE fish
     implicit none
     TYPE (fishworkspace) :: w
  Local Variables
C-----
     INTEGER :: IFLG, NP, N, MP, M, IDIMY, I, J, IERROR
     REAL , DIMENSION(105) :: AN, BN, CN
     REAL , DIMENSION (75) :: S
     REAL , DIMENSION (105) :: T
REAL::DELTAS, DELTAT, HDS, TDS, TEMP1, TEMP2, TEMP3, HDT, TDT, ER
R, Z
     COMPLEX , DIMENSION (75,105) :: Y
     COMPLEX, DIMENSION (75) :: AM, BM, CM
     IFLG = 0
     NP = 1
     N = 63
     MP = 1
     M = 50
     IDIMY = 75
C
С
    GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING THE
C COEFFICIENTS AND THE ARRAY Y.
```

```
DELTAS = 1./FLOAT(M + 1)
      DO I = 1, M
         S(I) = FLOAT(I)*DELTAS
      END DO
      DELTAT = 1./FLOAT(N + 1)
      DO J = 1, N
         T(J) = FLOAT(J)*DELTAT
      END DO
С
С
      COMPUTE THE COEFFICIENTS AM, BM, CM CORRESPONDING TO
THE S DIRECTION
C
      HDS = DELTAS/2.
      TDS = DELTAS + DELTAS
      DO I = 1, M
         TEMP1 = 1./(S(I)*TDS)
         TEMP2 = 1./((S(I)-HDS)*TDS)
         TEMP3 = 1./((S(I) + HDS) * TDS)
         AM(I) = CMPLX(TEMP1*TEMP2, 0.)
         CM(I) = CMPLX(TEMP1*TEMP3, 0.)
         BM(I) = (-(AM(I) + CM(I))) - (0.,1.)
      END DO
С
      COMPUTE THE COEFFICIENTS AN, BN, CN CORRESPONDING TO
THE T DIRECTION
      HDT = DELTAT/2.
      TDT = DELTAT + DELTAT
      DO 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))
      END DO
C
С
      COMPUTE RIGHT SIDE OF EQUATION
С
      DO J = 1, N
         Y(:M,J) = 3.75*S(:M)*T(J)*(S(:M)**4+T(J)**4) -
(0.,1.)*(S(:M)*T
     1 (J))**5
```

```
END DO
С
C
      THE NONZERO BOUNDARY CONDITIONS ENTER THE LINEAR
SYSTEM VIA
      THE RIGHT SIDE Y(I, J). IF THE EQUATIONS (3) GIVEN
ABOVE
      ARE EVALUATED AT I=M AND J=1,..., N THEN THE TERM
CM(M) *U(M+1,J)
      IS KNOWN FROM THE BOUNDARY CONDITION TO BE
CM(M) *T(J) **5.
      THEREFORE THIS TERM CAN BE INCLUDED IN THE RIGHT
SIDE Y(M,J).
      THE SAME ANALYSIS APPLIES AT J=N AND I=1,..,M.
NOTE THAT THE
      CORNER AT J=N, I=M INCLUDES CONTRIBUTIONS FROM BOTH
BOUNDARIES.
C
      Y(M, :N) = Y(M, :N) - CM(M) *T(:N) **5
      Y(:M,N) = Y(:M,N) - CN(N)*S(:M)**5
      CALL
CBLKTRI (IFLG, NP, N, AN, BN, CN, MP, M, AM, BM, CM, IDIMY, Y, IERROR,
W)
      IFLG = IFLG + 1
      DO WHILE (IFLG - 1 <= 0)
         CALL CBLKTRI (IFLG, NP, N, AN, BN, CN, MP, M,
AM, BM, CM, IDIMY
            , Y, IERROR, W)
         IFLG = IFLG + 1
      END DO
      ERR = 0.
      DO J = 1, N
         DO I = 1, M
            Z = CABS(Y(I, J) - (S(I) *T(J)) **5)
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
      WRITE (*, *) '
                       CBLKTRI TEST RUN *** '
      WRITE (*, *)
            Previous 64 bit floating point arithmetic
result '
```

```
WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.6457E-05'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.2737E-02'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
    release dynamically allocated work space
     CALL FISHFIN (W)
     STOP
     END PROGRAM TCBLKTRI
```

TCMGNBN

```
FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
С
                     the National Science Foundation
С
```

```
C
     PROGRAM TCMBNGN
     implicit none
  Local Variables
C-----
     INTEGER :: IDIMF, M, MP1, MPEROD, N, NPEROD, I, J,
TERROR
     REAL , DIMENSION (21) :: X
     REAL , DIMENSION (41) :: Y
     REAL :: DX, PI, DUM, DY, S, T, TSQ, T4, ERR
     COMPLEX , DIMENSION(22,40) :: F
     COMPLEX, DIMENSION(20) :: A, B, C
С
С
    PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE CMGNBN
TO SOLVE
    THE EQUATION
С
С
    (1+X)**2*(D/DX)(DU/DX) - 2(1+X)(DU/DX) +
(D/DY)(DU/DY)
С
С
           - SQRT(-1)*U = (3 - SQRT(-1))
1)) *(1+X) **4*SIN(Y) (1)
С
    ON THE RECTANGLE 0 .LT. X .LT. 1 AND -PI .LT. Y
.LT. PI
С
    WITH THE BOUNDARY CONDITIONS
C
C
    (DU/DX)(0,Y) = 4SIN(Y)
(2)
С
                           -PI .LE. Y .LE. PI
С
    U(1,Y) = 16SIN(Y)
(3)
С
    AND WITH U PERIODIC IN Y USING FINITE DIFFERENCES
С
ON A
    GRID WITH DELTAX (= DX) = 1/20 AND DELTAY (= DY) =
      TO SET UP THE FINITE DIFFERENCE EQUATIONS WE
DEFINE
C THE GRID POINTS
```

```
I=1,2,...,21
С
     X(I) = (I-1)DX
C
     Y(J) = -PI + (J-1)DY J=1, 2, ..., 41
С
С
C
     AND LET V(I, J) BE AN APPROXIMATION TO
U(X(I),Y(J)).
     NUMBERING THE GRID POINTS IN THIS FASHION GIVES
THE SET
     OF UNKNOWNS AS V(I,J) FOR I=1,2,...,20 AND
J=1,2,...,40.
     HENCE, IN THE PROGRAM M = 20 AND N = 40. AT THE
INTERIOR
      GRID POINT (X(I), Y(J)), WE REPLACE ALL DERIVATIVES
C
IN
     EQUATION (1) BY SECOND ORDER CENTRAL FINITE
DIFFERENCES,
     MULTIPLY BY DY**2, AND COLLECT COEFFICIENTS OF
V(I,J) TO
     GET THE FINITE DIFFERENCE EQUATION
С
C
     A(I)V(I-1,J) + B(I)V(I,J) + C(I)V(I+1,J)
C
С
     + V(I, J-1) - 2V(I, J) + V(I, J+1) = F(I, J)
(4)
С
С
     WHERE S = (DY/DX)**2, AND FOR I=2,3,...,19
С
С
     A(I) = (1+X(I))**2*S + (1+X(I))*S*DX
С
С
     B(I) = -2(1+X(I))**2*S - SQRT(-1)*DY**2
С
С
     C(I) = (1+X(I))**2*S - (1+X(I))*S*DX
С
С
     F(I,J) = (3 - SQRT(-
1))*(1+X(I))**4*DY**2*SIN(Y(J))
С
               FOR J=1,2,...,40.
С
         TO OBTAIN EQUATIONS FOR I = 1, WE REPLACE THE
С
C
     DERIVATIVE IN EQUATION (2) BY A SECOND ORDER
CENTRAL
      FINITE DIFFERENCE APPROXIMATION, USE THIS EQUATION
C
TO
C ELIMINATE THE VIRTUAL UNKNOWN V(0,J) IN EQUATION
```

```
(4)
С
      AND ARRIVE AT THE EQUATION
С
C
      B(1)V(1,J) + C(1)V(2,J) + V(1,J-1) - 2V(1,J) +
V(1,J+1)
C
С
                        = F(1, J)
С
С
      WHERE
С
С
      B(1) = -2S - SQRT(-1)*DY**2 , C(1) = 2S
С
      F(1,J) = (11-SQRT(-1)+8/DX)*DY**2*SIN(Y(J)),
C
J=1,2,...,40.
С
С
      FOR COMPLETENESS, WE SET A(1) = 0.
С
         TO OBTAIN EQUATIONS FOR I = 20, WE INCORPORATE
С
      EQUATION (3) INTO EQUATION (4) BY SETTING
С
С
      V(21,J) = 16SIN(Y(J))
С
С
     AND ARRIVE AT THE EQUATION
С
С
      A(20)V(19,J) + B(20)V(20,J)
С
С
      + V(20, J-1) - 2V(20, J) + V(20, J+1) = F(20, J)
С
С
      WHERE
С
C
      A(20) = (1+X(20))**2*S + (1+X(20))*S*DX
С
С
      B(20) = -2*(1+X(20))**2*S - SQRT(-1)*DY**2
С
C
      F(20,J) = ((3-SQRT(-1))*(1+X(20))**4*DY**2 -
16(1+X(20))**2*S
С
                 + 16(1+X(20))*S*DX)*SIN(Y(J))
С
С
                     FOR J=1,2,...,40.
С
С
      FOR COMPLETENESS, WE SET C(20) = 0. HENCE, IN THE
      PROGRAM MPEROD = 1.
C
С
         THE PERIODICITY CONDITION ON U GIVES THE
CONDITIONS
```

```
V(I,0) = V(I,40) AND V(I,41) = V(I,1) FOR
I=1,2,...,20.
С
      HENCE, IN THE PROGRAM NPEROD = 0.
С
С
С
           THE EXACT SOLUTION TO THIS PROBLEM IS
С
С
                   U(X,Y) = (1+X)**4*SIN(Y).
\mathsf{C}
С
С
      FROM THE DIMENSION STATEMENT WE GET THAT IDIMF =
22
C
      IDIMF = 22
     M = 20
     MP1 = M + 1
     MPEROD = 1
     DX = 0.05
     N = 40
      NPEROD = 0
     PI = 4.0*atan(1.0)
     DY = PI/20.
C
С
      GENERATE GRID POINTS FOR LATER USE.
С
      DO I = 1, MP1
         X(I) = FLOAT(I - 1)*DX
      END DO
      DO J = 1, N
         Y(J) = (-PI) + FLOAT(J - 1)*DY
      END DO
C
С
     GENERATE COEFFICIENTS.
C
      S = (DY/DX) **2
      DO I = 2, 19
         T = 1. + X(I)
         TSQ = T**2
         A(I) = CMPLX((TSQ + T*DX)*S, 0.)
         B(I) = (-2.*TSQ*S) - (0.,1.)*DY**2
         C(I) = CMPLX((TSQ - T*DX)*S, 0.)
      END DO
      A(1) = (0.,0.)
      B(1) = (-2.*S) - (0.,1.)*DY**2
```

```
C(1) = CMPLX(2.*S, 0.)
      B(20) = (-2.*S*(1. + X(20))**2) - (0.,1.)*DY**2
     A(20) = CMPLX(S*(1. +
X(20))**2+(1.+X(20))*DX*S,0.)
     C(20) = (0.,0.)
C
С
     GENERATE RIGHT SIDE.
С
     DO I = 2, 19
         DO J = 1, N
            F(I,J) = (3.,-1.)*(1. +
X(I))**4*DY**2*SIN(Y(J))
         END DO
      END DO
      T = 1. + X(20)
     TSQ = T**2
     T4 = TSO**2
     DO J = 1, N
         F(1,J) = ((11.,-1.) + 8./DX)*DY**2*SIN(Y(J))
         F(20,J) = ((3.,-1.)*T4*DY**2-
16.*TSQ*S+16.*T*S*DX)*SIN(Y(J))
      END DO
      CALL CMGNBN (NPEROD, N, MPEROD, M, A, B, C, IDIMF,
F, IERROR)
C
      COMPUTE DISCRETIAZATION ERROR. THE EXACT SOLUTION
IS
C
С
             U(X,Y) = (1+X)**4*SIN(Y).
С
      ERR = 0.
      DO I = 1, M
         DO J = 1, N
            T = CABS(F(I, J) - (1.+X(I)) **4*SIN(Y(J)))
            ERR = AMAX1 (T, ERR)
         END DO
      END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
      WRITE (*, *) ' CMGNBN TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
```

```
result '

WRITE (*, *) ' IERROR = 0, Discretization

Error = 9.1620E-3'

WRITE (*, *)

1 ' Previous 32 bit floating point arithmetic

result '

WRITE (*, *) ' IERROR = 0, Discretization

Error = 9.1801E-3'

WRITE (*, *) ' The output from your computer

is: '

WRITE (*, *) ' IERROR = ', IERROR, '

Discretization Error = ',

1 ERR

STOP

END PROGRAM TCMBNGN
```

TGENBUN

```
FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
С
                     the National Science Foundation
```

```
С
      PROGRAM TGENBUN
     implicit none
  Local Variables
      INTEGER :: IDIMF, M, MP1, MPEROD, N, NPEROD, I, J,
IERROR
     REAL , DIMENSION (22,40) :: F
     REAL , DIMENSION (20) :: A, B, C
     REAL , DIMENSION (21) :: X
     REAL , DIMENSION (41) :: Y
     REAL :: DX, PI, DY, S, T, TSQ, T4, ERR
С
     FROM THE DIMENSION STATEMENT WE GET THAT IDIMF =
22
C
     IDIMF = 22
     M = 20
     MP1 = M + 1
     MPEROD = 1
     DX = 0.05
     N = 40
     NPEROD = 0
     PI = 4.0*ATAN(1.0)
     DY = PI/20.
C
С
     GENERATE GRID POINTS FOR LATER USE.
     DO I = 1, MP1
        X(I) = FLOAT(I - 1)*DX
      END DO
      DO J = 1, N
         Y(J) = (-PI) + FLOAT(J - 1)*DY
     END DO
C
С
     GENERATE COEFFICIENTS.
С
     S = (DY/DX) **2
```

```
DO I = 2, 19
         T = 1. + X(I)
         TSO = T**2
         A(I) = (TSO + T*DX)*S
         B(I) = -2.*TSQ*S
         C(I) = (TSQ - T*DX)*S
      END DO
      A(1) = 0.
      B(1) = -2.*S
      C(1) = -B(1)
      B(20) = -2.*S*(1. + X(20))**2
      A(20) = (-B(20)/2.) + (1. + X(20))*DX*S
      C(20) = 0.
С
\mathsf{C}
      GENERATE RIGHT SIDE.
С
      DO I = 2, 19
         DO J = 1, N
            F(I,J) = 3.*(1. + X(I))**4*DY**2*SIN(Y(J))
         END DO
      END DO
      T = 1. + X(20)
      TSO = T**2
      T4 = TSQ**2
      DO J = 1, N
         F(1,J) = (11. + 8./DX)*DY**2*SIN(Y(J))
         F(20,J) = (3.*T4*DY**2 - 16.*TSQ*S +
16.*T*S*DX)*SIN(Y(J))
      END DO
      CALL GENBUN (NPEROD, N, MPEROD, M, A, B, C, IDIMF,
F, IERROR)
C
С
      COMPUTE DISCRETIAZATION ERROR. THE EXACT SOLUTION
IS
С
С
            U(X,Y) = (1+X)**4*SIN(Y).
С
      ERR = 0.
      DO I = 1, M
         DO J = 1, N
            T = ABS(F(I, J) - (1.+X(I)) **4*SIN(Y(J)))
            ERR = AMAX1 (T, ERR)
         END DO
      END DO
```

```
! Print earlier output from platforms with 32 and 64
bit floating point
    arithemtic followed by the output from this
computer
     WRITE (*, *) ' GENBUN TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 9.6406E-3'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 9.6556E-3'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
     STOP
     END PROGRAM TGENBUN
```

THSTCRT

```
all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
```

```
C
                   the National Science Foundation
С
C
     PROGRAM THSTCRT
     implicit none
C Local Variables
C-----
     INTEGER :: IDIMF, M, MBDCND, N, NBDCND, I, J,
IERROR
     REAL , DIMENSION (50,53) :: F
     REAL , DIMENSION (53) :: BDA, BDB
     REAL , DIMENSION (48) :: X
     REAL , DIMENSION (53) :: Y
REAL::A,B,DX,C,D,DY,ELMBDA,PI,PISQ,T,BDC,BDD,PERTRB,ERR
С
С
     FROM THE DIMENSION STATEMENT WE GET IDIMF = 50.
C
     IDIMF = 50
     A = 1.
     B = 3.
     M = 48
     DX = (B - A)/FLOAT(M)
     MBDCND = 2
     C = -1.
     D = 1.
     N = 53
     DY = (D - C)/FLOAT(N)
     NBDCND = 0
     ELMBDA = -2.
С
С
     AUXILIARY QUANTITIES
C
     PI = 4.0*ATAN(1.0)
     PISQ = PI*PI
С
```

```
C GENERATE AND STORE GRID POINTS FOR COMPUTATION OF
BOUNDARY DATA
С
     AND THE RIGHT SIDE OF THE HELMHOLTZ EQUATION.
C
      DO I = 1, M
         X(I) = A + (FLOAT(I) - 0.5)*DX
      END DO
      DO J = 1, N
         Y(J) = C + (FLOAT(J) - 0.5)*DY
      END DO
С
С
     GENERATE BOUNDARY DATA.
C
      DO J = 1, N
        BDA(J) = 0.
         BDB(J) = -PI*COS(PI*Y(J))
     END DO
C
С
     BDC AND BDD ARE DUMMY ARGUMENTS IN THIS EXAMPLE.
С
С
     GENERATE RIGHT SIDE OF EQUATION.
C
     T = -2.*(PISO + 1.)
     DO I = 1, M
         DO J = 1, N
            F(I,J) = T*SIN(PI*X(I))*COS(PI*Y(J))
         END DO
      END DO
      CALL HSTCRT (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
C
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
С
С
                U(X,Y) = SIN(PI*X)*COS(PI*Y).
С
      ERR = 0.
      DO I = 1, M
         DO J = 1, N
            T = ABS(F(I,J)-SIN(PI*X(I))*COS(PI*Y(J)))
            ERR = AMAX1 (T, ERR)
         END DO
      END DO
```

```
! Print earlier output from platforms with 32 and 64
bit floating point
    arithemtic followed by the output from this
computer
     WRITE (*, *) ' HSTCRT TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.2600E-3'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.2586E-3'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
     STOP
     END PROGRAM THSTCRT
```

THSTCSP

```
all rights reserved
С
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
```

```
C
                    the National Science Foundation
С
С
      PROGRAM THSTCSP
     USE fish
     implicit none
     TYPE (fishworkspace) :: w
   Local Variables
      INTEGER :: IDIMF, M, MBDCND, I, N, NBDCND, J,
INTL, IERROR
     REAL , DIMENSION (47,16) :: F
     REAL , DIMENSION (45) :: BDD, THETA
     REAL , DIMENSION (15) :: R
     REAL , DIMENSION (45) :: COST
      REAL :: A, B, DT, C, D, DR, ELMBDA, BDA, BDB, BDC,
PERTRB, ERR, Z
C-----
C
     NOTE THAT FROM DIMENSION STATEMENT WE GET THAT
IDIMF = 47
     IDIMF = 47
     A = 0.
     B = 4.0*ATAN(1.0)
С
     NOTE THAT B IS SET TO PI USING THE FUNCTION PIMACH
AS REQUIRED.
С
     M = 45
     MBDCND = 9
     DT = (B - A)/FLOAT(M)
С
C
     DEFINE GRID POINTS THETA(I) AND COS(THETA(I))
С
     DO I = 1, M
         THETA(I) = A + (FLOAT(I) - 0.5) *DT
```

```
COST(I) = COS(THETA(I))
      END DO
      C = 0.
      D = 1.
      N = 15
      NBDCND = 5
     DR = (D - C)/FLOAT(N)
C
C
     DEFINE GRID POINTS R(J)
С
      DO J = 1, N
         R(J) = C + (FLOAT(J) - 0.5)*DR
      END DO
C
С
      DEFINE BOUNDARY ARRAY BDD. BDA, BDB, AND BDC ARE
DUMMY
С
     VARIABLES IN THIS EXAMPLE.
С
      BDD(:M) = COST(:M) **4
     ELMBDA = 0.
С
С
     DEFINE RIGHT SIDE F
C
      DO I = 1, M
         F(I,:N) = 12.*(R(:N)*COST(I))**2
      END DO
      INTL = 0
      CALL HSTCSP (INTL, A, B, M, MBDCND, BDA, BDB, C,
D, N, NBDCND, BDC
     1 , BDD, ELMBDA, F, IDIMF, PERTRB, IERROR, W)
C
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
C
С
      U(THETA,R) = (R*COS(THETA))**4
С
      ERR = 0.
      DO I = 1, M
         DO J = 1, N
            Z = ABS(F(I,J) - (R(J) *COST(I)) **4)
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
     Print earlier output from platforms with 32 and 64
```

```
bit floating point
     arithemtic followed by the output from this
computer
     in this example (contrast with blktri and sepeli)
the extra precision
    does not reduce the discretization error
     WRITE (*, *) ' HSTCSP TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 5.5843E-3'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 5.5845E-3'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 FRR
    release work space allocated by hstcsp (intl=0
call)
     CALL FISHFIN (W)
     STOP
     END PROGRAM THSTCSP
```

THSTCYL

```
С
              University Corporation for Atmospheric
Research
                              all rights reserved
С
С
С
                            FISHPACK90 version 1.1
С
                         A Package of Fortran 77 and 90
С
С
*
С
                        Subroutines and Example Programs
С
                       for Modeling Geophysical Processes
\mathsf{C}
С
С
                                      by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                      of
С
                 the National Center for Atmospheric
С
Research
С
                        Boulder, Colorado (80307)
```

```
U.S.A.
С
*
С
                       which is sponsored by
С
С
                  the National Science Foundation
C
     PROGRAM THSTCYL
     implicit none
C-----
  Local Variables
C-----
     INTEGER :: IDIMF, M, MBDCND, N, NBDCND, I, J,
IERROR
     REAL , DIMENSION (51,52) :: F
     REAL , DIMENSION (52) :: BDB
     REAL , DIMENSION (50) :: BDC, BDD, R
     REAL , DIMENSION (52) :: Z
     REAL :: A, B, C, D, ELMBDA, BDA, PERTRB, X, ERR
C
     PROGRAM TO ILLUSTRATE THE USE OF HSTCYL TO SOLVE
THE EQUATION
    (1/R) (D/DR) (R*DU/DR) + (D/DZ) (DU/DZ) =
(2*R*Z)**2*(4*Z**2 + 3*R**2)
C
С
    ON THE RECTANGLE 0 .LT. R .LT. 1 , 0 .LT. Z .LT. 1
WITH THE
С
    BOUNDARY CONDITIONS
C
С
    (DU/DR)(1,Z) = 4*Z**2 FOR 0 .LE. Z .LE. 1
С
C
     AND
С
     (DU/DZ)(R,0) = 0 AND (DU/DZ)(R,1) = 4*R**2 FOR 0
.LE. R .LE. 1 .
```

```
C
С
      THE SOLUTION TO THIS PROBLEM IS NOT UNIQUE. IT IS
Α
     ONE-PARAMETER FAMILY OF SOLUTIONS GIVEN BY
С
С
С
             U(R,Z) = (R*Z)**4 + ARBITRARY CONSTANT.
С
      THE R-INTERVAL WILL CONTAIN 50 UNKNOWNS AND THE Z-
С
INTERVAL WILL
      CONTAIN 52 UNKNOWNS.
С
С
С
     FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.
С
     IDIMF = 51
     A = 0.
     B = 1.
     M = 50
     MBDCND = 6
     C = 0.
     D = 1.
     N = 52
     NBDCND = 3
     ELMBDA = 0.
C
     GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
C
COMPUTING
      BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON
EQUATION.
С
      DO I = 1, M
        R(I) = (FLOAT(I) - 0.5)/50.
      END DO
      DO J = 1, N
         Z(J) = (FLOAT(J) - 0.5)/52.
     END DO
C
С
     GENERATE BOUNDARY DATA.
С
     BDB(:N) = 4.*Z(:N)**4
С
С
     GENERATE BOUNDARY DATA.
С
     BDC(:M) = 0.
```

```
BDD(:M) = 4.*R(:M)**4
С
С
     BDA IS A DUMMY VARIABLE.
С
С
     GENERATE RIGHT SIDE OF EQUATION.
C
     DO I = 1, M
        F(I,:N) =
4.*R(I)**2*Z(:N)**2*(4.*Z(:N)**2+3.*R(I)**2)
     END DO
     CALL HSTCYL (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
C
С
     COMPUTE DISCRETIZATION ERROR BY MINIMIZING OVER
ALL A THE FUNCTION
     NORM(F(I,J) - A*1 - U(R(I),Z(J))). THE EXACT
SOLUTION IS
                U(R,Z) = (R*Z)**4 + ARBITRARY CONSTANT.
C
С
     X = 0.
     DO I = 1, M
        X = X + SUM(F(I,:N) - (R(I)*Z(:N))**4)
     END DO
     X = X/FLOAT(M*N)
     F(:M,:N) = F(:M,:N) - X
     ERR = 0.
     DO I = 1, M
         DO J = 1, N
           X = ABS(F(I,J) - (R(I)*Z(J))**4)
            ERR = AMAX1(X, ERR)
         END DO
     END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
     WRITE (*, *) ' HSTCYL TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, PERTRB =-4.4311E-4'
     WRITE (*, *) ' Discretization Error = 7.5280E-5
```

```
WRITE (*, *)

1 ' Previous 32 bit floating point arithmetic

result '

WRITE (*, *) ' IERROR = 0, PERTRB =-4.4321E-4'

WRITE (*, *) ' Discretization Error = 7.3557E-

5'

WRITE (*, *) ' The output from your computer

is: '

WRITE (*, *) ' IERROR =', IERROR, ' PERTRB = ',

PERTRB

WRITE (*, *) ' Discretization Error = ', ERR

STOP

END PROGRAM THSTCYL
```

THSTPLR

```
С
   file thstplr.f
С
   С
С
C
                  copyright (c) 2005 by UCAR
С
С
          University Corporation for Atmospheric
Research
                     all rights reserved
С
С
С
                   FISHPACK90 version 1.1
*
С
```

```
A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
                                    by
С
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                     of
                the National Center for Atmospheric
С
Research
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
                     the National Science Foundation
С
```

```
PROGRAM THSTPLR
     implicit none
C-----
  Local Variables
     INTEGER :: IDIMF, M, MBDCND, N, NBDCND, I, J,
IERROR
     REAL , DIMENSION (51,50) :: F
     REAL , DIMENSION (48) :: BDB
     REAL , DIMENSION (50) :: BDC, BDD, R
     REAL , DIMENSION (48) :: THETA
     REAL :: A, B, C, PI, D, ELMBDA, BDA, PERTRB, ERR,
С
    FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.
С
     IDIMF = 51
     A = 0.
     B = 1.
     M = 50
     MBDCND = 5
     C = 0.
     PI = 4.0*ATAN(1.0)
     D = PI/2.
     N = 48
     NBDCND = 3
     ELMBDA = 0.
C
С
     GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING
     BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON
EQUATION.
     DO I = 1, M
        R(I) = (FLOAT(I) - 0.5)/50.
     END DO
     DO J = 1, N
        THETA(J) = (FLOAT(J) - 0.5)*PI/96.
     END DO
C
С
     GENERATE BOUNDARY DATA.
С
    DO J = 1, N
```

```
BDB(J) = 1. - COS(4.*THETA(J))
      END DO
C
С
      GENERATE BOUNDARY DATA.
С
      BDC(:M) = 0.
      BDD(:M) = 0.
С
С
     BDA IS A DUMMY VARIABLE.
С
С
C
     GENERATE RIGHT SIDE OF EQUATION.
C
      DO I = 1, M
        F(I,:N) = 16.*R(I)**2
      END DO
      CALL HSTPLR (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
С
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
TS
C
                 U(R, THETA) = R**4*(1 - COS(4*THETA))
С
С
      ERR = 0.
      DO I = 1, M
         DO J = 1, N
            Z = ABS(F(I, J) - R(I) **4*(1.-
COS(4.*THETA(J))))
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
     WRITE (*, *) ' HSTPLR TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
      WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.1303E-3'
     WRITE (*, *)
```

```
1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 1.1300E-3'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR = ', IERROR, '
Discretization Error = ',
     1 ERR
     STOP
     END PROGRAM THSTPLR
```

THSTSSP

```
С
С
    file thstssp.f
С
С
С
                        copyright (c) 2005 by UCAR
С
C
             University Corporation for Atmospheric
Research
С
С
                            all rights reserved
С
С
                          FISHPACK90 version 1.1
С
*
С
                       A Package of Fortran 77 and 90
```

```
С
                       Subroutines and Example Programs
С
С
С
                      for Modeling Geophysical Processes
С
С
                                     by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
С
U.S.A.
С
                          which is sponsored by
С
С
С
                    the National Science Foundation
С
С
      PROGRAM THSTSSP
      implicit none
```

```
C Local Variables
C-----
     INTEGER :: M, MBDCND, N, NBDCND, IDIMF, I, J,
IERROR
     REAL , DIMENSION (18,72) :: F
     REAL , DIMENSION (72) :: BDB
     REAL , DIMENSION (18) :: SINT
     REAL , DIMENSION (72) :: SINP
REAL::PI,A,B,C,D,ELMBDA,DTHETA,DPHI,BDA,BDC,BDD,PERTRB,E
RR, Z
C
С
     THE VALUE OF IDIMF IS THE FIRST DIMENSION OF F.
C
     PI = 4.0*ATAN(1.0)
     A = 0.
     B = PI/2.
     M = 18
     MBDCND = 6
     C = 0.
     D = 2.*PI
     N = 72
     NBDCND = 0
     ELMBDA = 0.
     IDIMF = 18
С
С
    GENERATE SINES FOR USE IN SUBSEQUENT COMPUTATIONS
С
     DTHETA = B/FLOAT(M)
     DO I = 1, M
        SINT(I) = SIN((FLOAT(I) - 0.5)*DTHETA)
     END DO
     DPHI = D/FLOAT(N)
     DO J = 1, N
        SINP(J) = SIN((FLOAT(J) - 0.5)*DPHI)
     END DO
С
C
     COMPUTE RIGHT SIDE OF EQUATION AND STORE IN F
C
     DO J = 1, N
        F(:M,J) = 2. - 6.*(SINT(:M)*SINP(J))**2
     END DO
```

```
С
С
      STORE DERIVATIVE DATA AT THE EQUATOR
C
     BDB(:N) = 0.
С
С
     BDA, BDC, AND BDD ARE DUMMY VARIABLES.
C
     CALL HSTSSP (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
С
С
     COMPUTE DISCRETIZATION ERROR. SINCE PROBLEM IS
SINGULAR, THE
      SOLUTION MUST BE NORMALIZED.
C
С
     ERR = 0.
     DO J = 1, N
        DO I = 1, M
           Z = ABS(F(I,J) - (SINT(I) *SINP(J)) **2-F(1,1))
           ERR = AMAX1(Z, ERR)
        END DO
     END DO
     Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
     WRITE (*, *) ' HSTSSP TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, PERTRB = 6.35830E-
4 '
     WRITE (*, *) ' discretization error = 3.37523E-
3'
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, PERTRB = 6.35919E-
4 '
     WRITE (*, *) ' discretization error = 3.38144E-
31
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, ' PERTRB = ',
```

```
PERTRB

WRITE (*, *) ' discretization error = ', ERR

STOP

END PROGRAM THSTSSP
```

THW3CRT

```
C
С
   file thw3crt.f
С
С
                       copyright (c) 2005 by UCAR
С
С
           University Corporation for Atmospheric
Research
С
                           all rights reserved
С
                         FISHPACK90 version 1.1
С
С
                       A Package of Fortran 77 and 90
С
С
*
С
                      Subroutines and Example Programs
*
С
```

```
for Modeling Geophysical Processes
С
С
                                   by
С
С
              John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                   of
С
               the National Center for Atmospheric
Research
С
                      Boulder, Colorado (80307)
U.S.A.
C
С
                         which is sponsored by
С
                    the National Science Foundation
C
С
     PROGRAM THW3CRT
     implicit none
  Local Variables
      INTEGER :: LBDCND, MBDCND, NBDCND, L, M, N, LDIMF,
MDIMF, LP1, I,
     1 MP1, J, NP1, K, IERROR
     REAL , DIMENSION(11,41,16) :: F
```

```
REAL , DIMENSION(11,41) :: BDZF
     REAL , DIMENSION(11) :: X
     REAL , DIMENSION (41) :: Y
     REAL , DIMENSION (16) :: Z
     REAL :: ELMBDA, XS, XF, YS, PI, YF, ZS, ZF, DX,
DY, DZ, BDXS, BDXF
    1 , BDYS, BDYF, BDZS, PERTRB, ERR, T
C-----
C
        FROM THE DESCRIPTION OF THE PROBLEM GIVEN
C
ABOVE, WE DEFINE
     THE FOLLOWING QUANTITIES
С
     ELMBDA = -3.
     XS = 0.
     XF = 1.
     LBDCND = 1
     YS = 0.
     PI = 4.0*ATAN(1.0)
     YF = 2.*PI
     MBDCND = 0
     ZS = 0.
     ZF = PI/2.
     NBDCND = 2
     L = 10
     M = 40
     N = 15
С
С
     FROM THE DIMENSION STATEMENT ABOVE WE DEFINE
C
     LDIMF = 11
     MDIMF = 41
C
С
     WE DEFINE THE GRID POINTS FOR LATER USE.
С
     LP1 = L + 1
     DX = (XF - XS)/FLOAT(L)
     DO I = 1, LP1
        X(I) = XS + FLOAT(I - 1)*DX
     END DO
     MP1 = M + 1
     DY = (YF - YS)/FLOAT(M)
     DO J = 1, MP1
```

```
Y(J) = YS + FLOAT(J - 1)*DY
      END DO
      NP1 = N + 1
      DZ = (ZF - ZS)/FLOAT(N)
      DO K = 1, NP1
         Z(K) = ZS + FLOAT(K - 1)*DZ
      END DO
C
\mathsf{C}
      WE DEFINE THE ARRAY OF DERIVATIVE BOUNDARY VALUES.
С
      DO I = 1, LP1
         DO J = 1, MP1
            BDZF(I,J) = -X(I) **4*SIN(Y(J))
         END DO
      END DO
C
      NOTE THAT FOR THIS EXAMPLE ALL OTHER BOUNDARY
ARRAYS ARE
      DUMMY VARIABLES.
      WE DEFINE THE FUNCTION BOUNDARY VALUES IN THE F
ARRAY.
C
      DO J = 1, MP1
         DO K = 1, NP1
            F(1, J, K) = 0.
            F(LP1, J, K) = SIN(Y(J))*COS(Z(K))
         END DO
      END DO
      DO I = 1, LP1
         DO J = 1, MP1
            F(I,J,1) = X(I) **4*SIN(Y(J))
         END DO
      END DO
C
С
      WE NOW DEFINE THE VALUES OF THE RIGHT SIDE OF THE
HELMHOLTZ
С
      EQUATION.
C
      DO I = 2, L
         DO J = 1, MP1
            DO K = 2, NP1
               F(I,J,K) = 4.*X(I)**2*(3. -
X(I)**2)*SIN(Y(J))*COS(Z(K))
            END DO
```

```
END DO
     END DO
C
     CALL HW3CRT TO GENERATE AND SOLVE THE FINITE
DIFFERENCE EQUATION.
     CALL HW3CRT (XS, XF, L, LBDCND, BDXS, BDXF, YS,
YF, M, MBDCND,
     1 BDYS, BDYF, ZS, ZF, N, NBDCND, BDZS, BDZF,
ELMBDA, LDIMF, MDIMF
     2 , F, PERTRB, IERROR)
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
С
TO THE
C
    PROBLEM IS
С
С
       U(X,Y,Z) = X^* 4^* SIN(Y) *COS(Z)
С
     ERR = 0.
     DO I = 1, LP1
        DO J = 1, MP1
           DO K = 1, NP1
              T = ABS(F(I, J, K) -
X(I)**4*SIN(Y(J))*COS(Z(K))
              ERR = AMAX1 (T, ERR)
           END DO
        END DO
     END DO
     Print earlier output from platforms with 32 and 64
bit floating point
     arithemtic followed by the output from this
computer
     WRITE (*, *) ' HW3CRT TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 9.6480E-3'
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 9.6480E-3'
     WRITE (*, *) ' The output from your computer
```

```
is: '
    WRITE (*, *) '    IERROR =', IERROR, '
Discretization Error = ',
    1    ERR
    STOP
    END PROGRAM THW3CRT
```

THWSCRT

```
C file thwscrt.f
С
С
С
С
                       copyright (c) 2005 by UCAR
С
             University Corporation for Atmospheric
Research
С
                            all rights reserved
С
С
                          FISHPACK90 version 1.1
С
С
                       A Package of Fortran 77 and 90
*
С
С
                      Subroutines and Example Programs
```

```
for Modeling Geophysical Processes
С
С
С
                                   by
С
С
              John Adams, Paul Swarztrauber and Roland
Sweet
С
                                   of
С
               the National Center for Atmospheric
Research
С
                      Boulder, Colorado (80307)
U.S.A.
С
С
                         which is sponsored by
С
С
                    the National Science Foundation
С
     PROGRAM THWSCRT
     implicit none
C Local Variables
      INTEGER :: IDIMF, M, MBDCND, N, NBDCND, MP1, NP1,
I, J, IERROR
```

```
REAL , DIMENSION (45,82) :: F
      REAL , DIMENSION(81) :: BDB, Y
      REAL , DIMENSION (41) :: X
REAL::A,B,C,D,ELMBDA,PI,DUM,PIBY2,PISQ,BDA,BDC,BDD,PERTR
B, ERR, Z
С
     FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.
С
      IDIMF = 45
     A = 0.
      B = 2.
     M = 40
      MBDCND = 2
      C = -1.
     D = 3.
      N = 80
      NBDCND = 0
     ELMBDA = -4.
С
C
     AUXILIARY QUANTITIES.
C
      PI = 4.0*ATAN(1.0)
      PIBY2 = PI/2.
      PISO = PI**2
      MP1 = M + 1
      NP1 = N + 1
C
С
      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING
      BOUNDARY DATA AND THE RIGHT SIDE OF THE HELMHOLTZ
EQUATION.
      DO I = 1, MP1
         X(I) = FLOAT(I - 1)/20.
      END DO
      DO J = 1, NP1
         Y(J) = (-1.) + FLOAT(J - 1)/20.
      END DO
C
С
      GENERATE BOUNDARY DATA.
С
     DO J = 1, NP1
```

```
BDB(J) = 4.*COS((Y(J)+1.)*PIBY2)
      END DO
C
      BDA, BDC, AND BDD ARE DUMMY VARIABLES.
С
С
      F(1,:NP1) = 0.
С
С
      GENERATE RIGHT SIDE OF EQUATION.
С
     DO I = 2, MP1
         DO J = 1, NP1
            F(I,J) = (2. - (4. +
PISQ/4.) *X(I) **2) *COS((Y(J)+1.) *PIBY2)
        END DO
      END DO
      CALL HWSCRT (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
C
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
С
                 U(X,Y) = X**2*COS((Y+1)*PIBY2)
C
     ERR = 0.
      DO I = 1, MP1
         DO J = 1, NP1
            Z = ABS(F(I, J) - X(I) **2*COS((Y(J) + 1.) *PIBY2))
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
     WRITE (*, *) ' HWSCRT TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
      WRITE (*, *) ' IERROR = 0, Discretization
Error = 5.36508-4
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
      WRITE (*, *) ' IERROR = 0, Discretization
```

```
Error = 4.9305E-4'
    WRITE (*, *) ' The output from your computer
is: '
    WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
    STOP
    END PROGRAM THWSCRT
```

THWSCSP

```
С
   file thwscsp.f
С
С
    *
С
                     copyright (c) 2005 by UCAR
C
С
           University Corporation for Atmospheric
Research
C
                          all rights reserved
С
С
                        FISHPACK90 version 1.1
С
*
С
                      A Package of Fortran 77 and 90
С
```

```
Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                    of
С
                the National Center for Atmospheric
Research
С
                       Boulder, Colorado (80307)
U.S.A.
                          which is sponsored by
С
С
С
                     the National Science Foundation
С
      PROGRAM THWSCSP
      USE fish
     implicit none
      TYPE (fishworkspace) :: w
```

```
C Local Variables
INTEGER::INTL, M, MBDCND, N, NBDCND, IDIMF, MP1, I, NP1, J, IERROR
      REAL , DIMENSION (48,33) :: F
      REAL , DIMENSION (33) :: BDTF
      REAL , DIMENSION (48) :: THETA
      REAL , DIMENSION(33) :: R
      REAL :: PI, DUM, TS, TF, RS, RF, ELMBDA, DTHETA,
DR, CI4, BDTS,
     1 BDRS, BDRF, PERTRB, ERR, Z, DPHI, SI
C
С
      PROGRAM TO ILLUSTRATE THE USE OF HWSCSP
С
С
     PI = 4.0*ATAN(1.0)
     INTL = 0
     TS = 0.
     TF = PI/2.
     M = 36
     MBDCND = 6
     RS = 0.
     RF = 1.
      N = 32
      NBDCND = 5
      ELMBDA = 0.
      IDIMF = 48
C
С
      GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING THE
C
      BOUNDARY DATA AND THE RIGHT SIDE OF THE EQUATION.
С
      MP1 = M + 1
      DTHETA = TF/FLOAT(M)
      DO I = 1, MP1
         THETA (I) = FLOAT (I -1) *DTHETA
      END DO
      NP1 = N + 1
      DR = 1./FLOAT(N)
      DO J = 1, NP1
        R(J) = FLOAT(J - 1)*DR
     END DO
С
```

```
GENERATE NORMAL DERIVATIVE DATA AT EQUATOR
С
      BDTF(:NP1) = 0.
C
С
      COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE
С
      DO I = 1, MP1
         F(I,N+1) = COS(THETA(I))**4
      END DO
С
С
      COMPUTE RIGHT SIDE OF EQUATION
С
      DO I = 1, MP1
         CI4 = 12.*COS(THETA(I))**2
         F(I,:N) = CI4*R(:N)**2
      END DO
      CALL HWSCSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF,
       NBDCND, BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB,
     1
IERROR, W)
С
C
      COMPUTE DISCRETIZATION ERROR
C
      ERR = 0.
      DO I = 1, MP1
         CI4 = COS(THETA(I))**4
         DO J = 1, N
            Z = ABS(F(I,J)-CI4*R(J)**4)
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
      WRITE (*, *) ' HWSCSP TEST RUN, EXAMPLE 1 *** '
      WRITE (*, *) ' Previous 64 bit floating point
arithmetic result '
      WRITE (*, *) ' ierror = 0'
      WRITE (*, *) ' discretization error = 7,9984E-4 '
      WRITE (*, *) ' Previous 32 bit floating point
arithmetic result '
      WRITE (*, *) ' ierror = 0'
      WRITE (*, *) ' discretization error = 7.9907E-4 '
```

```
WRITE (*, *) ' The output from your computer is: '
      WRITE (*, *) ' IERROR =', IERROR
      WRITE (*, *) ' Discretization Error =', ERR
C
С
      THE FOLLOWING PROGRAM ILLUSTRATES THE USE OF
HWSCSP TO SOLVE
      A THREE DIMENSIONAL PROBLEM WHICH HAS LONGITUDNAL
DEPENDENCE
C
      MBDCND = 2
      NBDCND = 1
     DPHI = PI/72.
      ELMBDA = -2.*(1. - COS(DPHI))/DPHI**2
C
С
     COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE
C
      DO I = 1, MP1
         F(I,N+1) = SIN(THETA(I))
      END DO
С
С
     COMPUTE RIGHT SIDE OF THE EQUATION
C
      F(:MP1,:N) = 0.
      CALL HWSCSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF,
RS, RF, N,
       NBDCND, BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB,
     1
IERROR, W)
C
      COMPUTE DISCRETIZATION ERROR (FOURIER
COEFFICIENTS)
      ERR = 0
      DO I = 1, MP1
         SI = SIN(THETA(I))
         DO J = 1, NP1
            Z = ABS(F(I,J)-R(J)*SI)
            ERR = AMAX1(Z, ERR)
         END DO
      END DO
C
      Print earlier output from platforms with 32 and 64
bit floating point
      arithemtic followed by the output from this
computer
```

```
WRITE (*, *) ' ******* '
     WRITE (*, *) ' ******* '
     WRITE (*, *) ' HWSCSP TEST RUN, EXAMPLE 2 *** '
     WRITE (*, *) ' Previous 64 bit floating point
arithmetic result '
     WRITE (*, *) ' ierror = 0'
     WRITE (*, *) ' discretization error = 5.8682E-5 '
     WRITE (*, *) ' Previous 32 bit floating point
arithmetic result '
     WRITE (*, *) ' ierror = 0'
     WRITE (*, *) ' discretization error = 5.9962E-5 '
     WRITE (*, *) ' The output from your computer is: '
     WRITE (*, *) ' IERROR =', IERROR
     WRITE (*, *) ' Discretization Error =', ERR
     release real and complex allocated work space
     CALL FISHFIN (W)
     STOP
     END PROGRAM THWSCSP
```

THWSCYL

```
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
                      for Modeling Geophysical Processes
С
С
*
С
                                    by
С
               John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                     of
С
                the National Center for Atmospheric
Research
                       Boulder, Colorado (80307)
U.S.A.
С
                          which is sponsored by
С
С
*
                     the National Science Foundation
```

```
С
     PROGRAM THWSCYL
     implicit none
  Local Variables
C-----
     INTEGER :: IDIMF, M, MBDCND, N, NBDCND, MP1, NP1,
I, J, IERROR
     REAL , DIMENSION (75,105) :: F
     REAL , DIMENSION (101) :: BDA, BDB
     REAL , DIMENSION (51) :: BDC, BDD, R
     REAL , DIMENSION(101) :: Z
     REAL :: A, B, C, D, ELMBDA, PERTRB, X, ERR
С
С
     FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.
C
     IDIMF = 75
     A = 0.
     B = 1.
     M = 50
     MBDCND = 6
     C = 0.
     D = 1.
     N = 100
     NBDCND = 3
     ELMBDA = 0.
С
C
    AUXILIARY QUANTITIES.
С
     MP1 = M + 1
     NP1 = N + 1
С
     GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING
     BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON
EOUATION.
С
DO I = 1, MP1
```

```
R(I) = FLOAT(I - 1)/50.
      END DO
      DO J = 1, NP1
         Z(J) = FLOAT(J - 1)/100.
      END DO
C
С
      GENERATE BOUNDARY DATA.
С
      BDB(:NP1) = 4.*Z(:NP1)**4
С
      GENERATE BOUNDARY DATA.
С
С
      BDC(:MP1) = 0.
      BDD(:MP1) = 4.*R(:MP1)**4
С
С
      BDA IS A DUMMY VARIABLE.
С
С
С
      GENERATE RIGHT SIDE OF EQUATION.
С
      DO I = 1, MP1
         F(I,:NP1) =
4.*R(I)**2*Z(:NP1)**2*(4.*Z(:NP1)**2+3.*R(I)**2)
      END DO
      CALL HWSCYL (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR)
C
      COMPUTE DISCRETIZATION ERROR BY MINIMIZING OVER
ALL A THE FUNCTION
     NORM(F(I, J) - A*1 - U(R(I), Z(J))). THE EXACT
SOLUTION IS
                 U(R,Z) = (R*Z)**4 + ARBITRARY CONSTANT.
C
С
      X = 0.
      DO I = 1, MP1
         X = X + SUM(F(I, :NP1) - (R(I) *Z(:NP1)) **4)
      END DO
      X = X/FLOAT(NP1*MP1)
      F(:MP1,:NP1) = F(:MP1,:NP1) - X
      ERR = 0.
      DO I = 1, MP1
         DO J = 1, NP1
            X = ABS(F(I,J) - (R(I)*Z(J))**4)
```

```
ERR = AMAX1(X, ERR)
        END DO
     END DO
     Print earlier output from platforms with 32 and 64
bit floating point
     arithemtic followed by the output from this
computer
     WRITE (*, *) ' HWSCYL TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, PERTRB = 2.2674E-4'
     WRITE (*, *) ' Discretization Error = 3.7367E-4
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, PERTRB = 2.26976-4'
     WRITE (*, *) ' Discretization Error = 3.5554E-
4 '
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, ' PERTRB = ',
PERTRB
     WRITE (*, *) ' Discretization Error = ', ERR
     STOP
     END PROGRAM THWSCYL
```

THWSPLR

```
С
              University Corporation for Atmospheric
Research
                              all rights reserved
С
С
С
                            FISHPACK90 version 1.1
С
                         A Package of Fortran 77 and 90
С
С
*
С
                        Subroutines and Example Programs
С
                       for Modeling Geophysical Processes
\mathsf{C}
С
С
                                      by
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
                                      of
С
                 the National Center for Atmospheric
С
Research
С
                        Boulder, Colorado (80307)
```

```
U.S.A.
С
*
С
                        which is sponsored by
С
С
                   the National Science Foundation
C
C
     PROGRAM THWSPLR
     implicit none
  Local Variables
C-----
     INTEGER :: IDIMF, M, MBDCND, N, NBDCND, MP1, NP1,
I, J, IERROR
     REAL , DIMENSION (100,50) :: F
     REAL , DIMENSION (51) :: BDC, BDD, R
     REAL , DIMENSION (49) :: THETA
     REAL :: A, B, C, PI, DUM, D, ELMBDA, BDA, BDB,
PERTRB, W, ERR, Z
         PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE
HWSPLR TO SOLVE
C
     THE EQUATION
C
    (1/R) (D/DR) (R*(DU/DR)) +
(1/R**2) (D/DTHETA) (DU/DTHETA) = 16*R**2
C
     ON THE QUARTER-DISK 0 .LT. R .LT. 1, 0 .LT. THETA
.LT. PI/2 WITH
     WITH THE BOUNDARY CONDITIONS
С
    U(1, THETA) = 1 - COS(4*THETA), 0 .LE. THETA .LE. 1
C
C
С
     AND
С
С
    (DU/DTHETA)(R,0) = (DU/DTHETA)(R,PI/2) = 0, 0
```

```
.LE. R .LE. 1.
С
С
     (NOTE THAT THE SOLUTION U IS UNSPECIFIED AT R =
0.)
С
           THE R-INTERVAL WILL BE DIVIDED INTO 50 PANELS
AND THE
      THETA-INTERVAL WILL BE DIVIDED INTO 48 PANELS.
C
С
С
С
     FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.
С
     IDIMF = 100
     A = 0.
     B = 1.
     M = 50
     MBDCND = 5
     C = 0.
     PI = 4.0*ATAN(1.0)
     D = PI/2.
     N = 48
     NBDCND = 3
     ELMBDA = 0.
C
С
     AUXILIARY QUANTITIES.
С
     MP1 = M + 1
     NP1 = N + 1
C
     GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF
COMPUTING
     BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON
EQUATION.
С
      DO I = 1, MP1
        R(I) = FLOAT(I - 1)/50.
      END DO
      DO J = 1, NP1
         THETA (J) = FLOAT (J - 1) *PI/96.
      END DO
C
С
     GENERATE BOUNDARY DATA.
С
     BDC(:MP1) = 0.
     BDD(:MP1) = 0.
```

```
С
С
     BDA AND BDB ARE DUMMY VARIABLES.
C
     DO J = 1, NP1
        F(MP1,J) = 1. - COS(4.*THETA(J))
     END DO
C
С
     GENERATE RIGHT SIDE OF EQUATION.
С
     DO I = 1, M
        F(I,:NP1) = 16.*R(I)**2
     END DO
     CALL HWSPLR (A, B, M, MBDCND, BDA, BDB, C, D, N,
NBDCND, BDC, BDD
     1 , ELMBDA, F, IDIMF, PERTRB, IERROR, W)
С
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
С
                U(R, THETA) = R^{**}4^{*}(1 - COS(4^{*}THETA))
С
     ERR = 0.
     DO I = 1, MP1
        DO J = 1, NP1
           Z = ABS(F(I,J)-R(I)**4*(1.-
COS (4.*THETA(J))))
           ERR = AMAX1(Z, ERR)
        END DO
     END DO
     WRITE (*, *) ' HWSPLR TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 6.19134E-4'
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 6.20723E-4'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
 1 ERR
```

```
STOP
END PROGRAM THWSPLR
```

THWSSSP

```
file thwsssp.f
С
                       copyright (c) 2005 by UCAR
С
С
             University Corporation for Atmospheric
Research
C
                            all rights reserved
С
С
                          FISHPACK90 version 1.1
С
                       A Package of Fortran 77 and 90
С
С
                       Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
```

```
С
                                   by
С
С
              John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                   of
С
               the National Center for Atmospheric
Research
                      Boulder, Colorado (80307)
U.S.A.
С
С
                         which is sponsored by
С
С
                    the National Science Foundation
С
C
С
С
      PROGRAM TO ILLUSTRATE THE USE OF HWSSSP
С
     PROGRAM THWSSSP
     implicit none
  Local Variables
      INTEGER :: M, MBDCND, N, NBDCND, IDIMF, MP1, I,
NP1, J, IERROR
     REAL , DIMENSION (19,73) :: F
     REAL , DIMENSION (73) :: BDTF
```

```
REAL , DIMENSION (19) :: SINT
      REAL , DIMENSION (73) :: SINP
      REAL :: PI, TS, TF, PS, PF, ELMBDA, DTHETA, DPHI,
BDTS, BDPS, BDPF
     1 , PERTRB, ERR, Z
      PI = 4.0*ATAN(1.0)
      TS = 0
      TF = PI/2.
      M = 18
      MBDCND = 6
      PS = 0
      PF = PI + PI
      N = 72
      NBDCND = 0
      ELMBDA = 0.
      IDIMF = 19
С
С
     GENERATE SINES FOR USE IN SUBSEQUENT COMPUTATIONS
С
      DTHETA = TF/FLOAT(M)
      MP1 = M + 1
      DO I = 1, MP1
         SINT(I) = SIN(FLOAT(I - 1)*DTHETA)
      END DO
      DPHI = (PI + PI) / FLOAT(N)
      NP1 = N + 1
      DO J = 1, NP1
         SINP(J) = SIN(FLOAT(J - 1)*DPHI)
      END DO
C
C
      COMPUTE RIGHT SIDE OF EQUATION AND STORE IN F
С
      DO J = 1, NP1
         F(:MP1,J) = 2. - 6.*(SINT(:MP1)*SINP(J))**2
      END DO
C
С
      STORE DERIVATIVE DATA AT THE EQUATOR
С
      BDTF(:NP1) = 0.
      CALL HWSSSP (TS, TF, M, MBDCND, BDTS, BDTF, PS,
PF, N, NBDCND,
    1 BDPS, BDPF, ELMBDA, F, IDIMF, PERTRB, IERROR)
```

```
С
     COMPUTE DISCRETIZATION ERROR. SINCE PROBLEM IS
SINGULAR, THE
C
     SOLUTION MUST BE NORMALIZED.
С
     ERR = 0
     DO J = 1, NP1
       DO I = 1, MP1
           Z = ABS(F(I, J) - (SINT(I) *SINP(J)) **2-F(1, 1))
           ERR = AMAX1(Z, ERR)
        END DO
     END DO
С
     WRITE (*, *) ' HWSSSP TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 3.38107E-3'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 3.3650E-3'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',
    1 ERR
     STOP
     END PROGRAM THWSSSP
```

TPOIS3D

```
С
                         copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
С
                              all rights reserved
С
С
                            FISHPACK90 version 1.1
С
                        A Package of Fortran 77 and 90
С
С
                        Subroutines and Example Programs
С
С
                      for Modeling Geophysical Processes
С
С
                                     by
С
С
               John Adams, Paul Swarztrauber and Roland
С
Sweet
С
С
                                     of
С
*
                the National Center for Atmospheric
```

```
Research
C
C
                       Boulder, Colorado (80307)
U.S.A.
C
С
                          which is sponsored by
С
С
                     the National Science Foundation
С
С
     PROGRAM TPOIS3D
     implicit none
  Local Variables
INTEGER::LDIMF, MDIMF, LPEROD, L, MPEROD, M, NPEROD, N, I, J, K, IE
RROR
     REAL , DIMENSION (32, 33, 10) :: F
     REAL , DIMENSION(10) :: A, B, C
     REAL , DIMENSION(30) :: X, Y
     REAL , DIMENSION(10) :: Z
     REAL :: PI, DX, C1, DY, C2, DZ, DZSQ, T, ERR
С
С
      FROM THE DIMENSION STATEMENT WE GET THAT LDIMF =
32, MDIMF = 33,
С
     LDIMF = 32
     MDIMF = 33
     PI = 4.0*ATAN(1.0)
     LPEROD = 0
     L = 30
     DX = 2.*PI/FLOAT(L)
     C1 = 1./DX**2
     MPEROD = 0
```

```
M = 30
      DY = 2.*PI/FLOAT(M)
      C2 = 1./DY**2
      NPEROD = 1
      N = 10
      DZ = 1./FLOAT(N)
      DZSQ = 1./DZ**2
C
С
     GENERATE GRID POINTS FOR LATER USE.
С
      DO I = 1, L
         X(I) = (-PI) + FLOAT(I - 1)*DX
      END DO
      DO J = 1, M
         Y(J) = (-PI) + FLOAT(J - 1)*DY
      END DO
С
С
      GENERATE COEFFICIENTS
С
     A(1) = 0.
     B(1) = -2.*DZSQ
     C(1) = -B(1)
      Z(1) = 0.
      DO K = 2, N
         Z(K) = FLOAT(K - 1)*DZ
         T = 1. + Z(K)
        A(K) = T^* 2^* DZSQ + T/DZ
         B(K) = -2.*T**2*DZSQ
         C(K) = T**2*DZSQ - T/DZ
     END DO
С
С
      GENERATE RIGHT SIDE OF EQUATION
С
      DO I = 1, L
         DO J = 1, M
            DO K = 2, N
               F(I,J,K) = 2.*SIN(X(I))*SIN(Y(J))*(1. +
Z(K)) **4
            END DO
         END DO
      END DO
      DO I = 1, L
         DO J = 1, L
            F(I,J,1) = (10. + 8./DZ)*SIN(X(I))*SIN(Y(J))
```

```
F(I,J,N) = F(I,J,N) -
C(N) *16.*SIN(X(I)) *SIN(Y(J))
        END DO
     END DO
     C(N) = 0.
C
С
     CALL POIS3D TO SOLVE EQUATIONS.
С
     CALL POIS3D (LPEROD, L, C1, MPEROD, M, C2, NPEROD,
N, A, B, C,
    1 LDIMF, MDIMF, F, IERROR)
С
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
С
С
              U(X,Y,Z) = SIN(X) *SIN(Y) * (1+Z) **4
С
     ERR = 0.
     DO I = 1, L
        DO J = 1, M
           DO K = 1, N
              T = ABS(F(I,J,K) -
SIN(X(I))*SIN(Y(J))*(1.+Z(K))**4)
              ERR = AMAX1 (T, ERR)
           END DO
         END DO
     END DO
     Print earlier output from platforms with 32 and 64
bit floating point
     arithemtic followed by the output from this
computer
     WRITE (*, *) ' POIS3D TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 2.93277E-2'
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 2.93390E-2'
     WRITE (*, *) ' The output from your computer
is: '
```

```
WRITE (*, *) ' IERROR =', IERROR, '
Discretization Error = ',

1 ERR
STOP
END PROGRAM TPOIS3D
```

TPOISTG

```
С
С
     file tpoistg.f
С
С
С
                       copyright (c) 2005 by UCAR
С
С
             University Corporation for Atmospheric
Research
С
                            all rights reserved
С
С
                          FISHPACK90 version 1.1
С
                       A Package of Fortran 77 and 90
С
С
С
                       Subroutines and Example Programs
*
С
```

```
for Modeling Geophysical Processes
С
С
С
                                   by
С
С
              John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                   of
С
               the National Center for Atmospheric
С
Research
                      Boulder, Colorado (80307)
U.S.A.
С
                         which is sponsored by
С
С
                    the National Science Foundation
С
С
     PROGRAM TPOISTG
     implicit none
  Local Variables
      INTEGER :: IDIMF, MPEROD, M, NPEROD, N, I, J,
IERROR
     REAL , DIMENSION(42,20) :: F
```

```
REAL, DIMENSION (40) :: A, B, C, X
      REAL , DIMENSION(20) :: Y
     REAL :: PI, DX, DY, S, ERR, T
С
С
      PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE POISTG
TO
С
     SOLVE THE EQUATION
С
С
     (1/\cos(X))(D/DX)(\cos(X)(DU/DX)) + (D/DY)(DU/DY) =
С
С
            2*Y**2*(6-Y**2)*SIN(X)
С
С
     ON THE RECTANGLE -PI/2 .LT. X .LT. PI/2 AND
С
      0 .LT. Y .LT. 1 WITH THE BOUNDARY CONDITIONS
C
     (DU/DX) (-PI/2,Y) = (DU/DX) (PI/2,Y) = 0 , 0 .LE. Y
С
.LE. 1 (2)
C
С
     U(X, 0) = 0
(3)
C
                                 -PI/2 .LE. X .LE. PI/2
С
     (DU/DY)(X,1) = 4SIN(X)
(4)
C
C
     USING FINITE DIFFERENCES ON A STAGGERED GRID WITH
     DELTAX (= DX) = PI/40 AND DELTAY (= DY) = 1/20.
С
C
        TO SET UP THE FINITE DIFFERENCE EQUATIONS WE
DEFINE
С
     THE GRID POINTS
C
     X(I) = -PI/2 + (I-0.5)DX
С
                                         I=1,2,...,40
С
                                          J=1,2,...,20
С
     Y(J) = (J-0.5) DY
С
     AND LET V(I, J) BE AN APPROXIMATION TO
C
U(X(I),Y(J)).
     NUMBERING THE GRID POINTS IN THIS FASHION GIVES
    OF UNKNOWNS AS V(I,J) FOR I=1,2,...,40 AND
J=1,2,...,20.
     HENCE, IN THE PROGRAM M = 40 AND N = 20. AT THE
INTERIOR
C GRID POINT (X(I),Y(J)), WE REPLACE ALL DERIVATIVES
```

```
IN
      EQUATION (1) BY SECOND ORDER CENTRAL FINITE
С
DIFFERENCES,
      MULTIPLY BY DY**2, AND COLLECT COEFFICIENTS OF
V(I,J) TO
      GET THE FINITE DIFFERENCE EQUATION
С
С
      A(I)V(I-1,J) + B(I)V(I,J) + C(I)V(I+1,J)
С
С
      + V(I, J-1) - 2V(I, J) + V(I, J+1) = F(I, J)
(5)
С
С
      WHERE S = (DY/DX)**2, AND FOR I=2,3,...,39
С
С
      A(I) = S*COS(X(I)-DX/2)
С
С
      B(I) = -S*(COS(X(I)-DX/2)+COS(X(I)+DX/2))
С
С
     C(I) = S*COS(X(I)+DX/2)
С
С
      F(I,J) = 2DY^{*}2^{Y}(J)^{*}2^{X}(6-Y(J)^{*}2)^{X}SIN(X(I))
J=1,2,...,19.
C
         TO OBTAIN EQUATIONS FOR I = 1, WE REPLACE
EQUATION (2)
C
      BY THE SECOND ORDER APPROXIMATION
C
С
      (V(1,J)-V(0,J))/DX = 0
С
С
      AND USE THIS EQUATION TO ELIMINATE V(0,J) IN
EQUATION (5)
C
      TO ARRIVE AT THE EQUATION
С
      B(1)V(1,J) + C(1)V(2,J) + V(1,J-1) - 2V(1,J) +
V(1,J+1)
С
С
                         = F(1, J)
С
С
      WHERE
С
С
      B(1) = -S*(COS(X(1)-DX/2)+COS(X(1)+DX/2))
С
\mathsf{C}
      C(1) = -B(1)
С
```

```
C FOR COMPLETENESS, WE SET A(1) = 0.
С
         TO OBTAIN EQUATIONS FOR I = 40, WE REPLACE THE
DERIVATIVE
      IN EQUATION (2) AT X=PI/2 IN A SIMILAR FASHION,
USE THIS
      EQUATION TO ELIMINATE THE VIRTUAL UNKNOWN V(41, J)
IN EQUATION
      (5) AND ARRIVE AT THE EQUATION
С
С
      A(40)V(39,J) + B(40)V(40,J)
С
С
     + V(40, J-1) - 2V(40, J) + V(40, J+1) = F(40, J)
С
С
      WHERE
С
C
      A(40) = -B(40) = -S*(COS(X(40) -
DX/2) + COS(X(40) + DX/2)
C
С
      FOR COMPLETENESS, WE SET C(40) = 0. HENCE, IN THE
      PROGRAM MPEROD = 1.
С
C
         FOR J = 1, WE REPLACE EQUATION (3) BY THE
SECOND ORDER
      APPROXIMATION
C
С
                  (V(I,0) + V(I,1))/2 = 0
С
С
С
      TO ARRIVE AT THE CONDITION
С
С
                 V(I,0) = -V(I,1) .
С
C
      FOR J = 20, WE REPLACE EQUATION (4) BY THE SECOND
ORDER
      APPROXIMATION
С
С
С
                  (V(I,21) - V(I,20))/DY = 4*SIN(X)
С
C
      AND COMBINE THIS EQUATION WITH EQUATION (5) TO
ARRIVE AT
      THE EQUATION
С
С
С
      A(I)V(I-1,20) + B(I)V(I,20) + C(I)V(I+1,20)
С
\mathsf{C}
      + V(I,19) - 2V(I,20) + V(I,21) = F(I,20)
С
```

```
C
      WHERE
С
C
      V(I,21) = V(I,20) \quad AND
C
С
      F(I,20) = 2*DY**2*Y(J)**2*(6-Y(J)**2)*SIN(X(I)) -
4*DY*SIN(X(I))
C
С
      HENCE, IN THE PROGRAM NPEROD = 2 .
С
         THE EXACT SOLUTION TO THIS PROBLEM IS
С
С
         U(X,Y) = Y^{**}4*COS(X) .
C
С
С
      FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF =
42
С
      IDIMF = 42
      MPEROD = 1
      M = 40
      PI = 4.0*ATAN(1.0)
      DX = PI/FLOAT(M)
      NPEROD = 2
      N = 20
      DY = 1./FLOAT(N)
С
С
      GENERATE AND STORE GRID POINTS FOR COMPUTATION.
С
      DO I = 1, M
         X(I) = (-PI/2.) + (FLOAT(I) - 0.5)*DX
      END DO
      DO J = 1, N
         Y(J) = (FLOAT(J) - 0.5)*DY
      END DO
С
      GENERATE COEFFICIENTS .
С
С
      S = (DY/DX) **2
      A(1) = 0.
      B(1) = -S*COS((-PI/2.) + DX)/COS(X(1))
      C(1) = -B(1)
      DO I = 2, M
         A(I) = S*COS(X(I)-DX/2.)/COS(X(I))
         C(I) = S*COS(X(I)+DX/2.)/COS(X(I))
         B(I) = -(A(I) + C(I))
```

```
END DO
     A(40) = -B(40)
     C(40) = 0.
C
С
     GENERATE RIGHT SIDE OF EQUATION.
С
     DO I = 1, M
        DO J = 1, N
           F(I,J) = 2.*DY**2*Y(J)**2*(6. -
Y(J) **2) *SIN(X(I))
        END DO
     END DO
     DO I = 1, M
        F(I,N) = F(I,N) - 4.*DY*SIN(X(I))
     END DO
      CALL POISTG (NPEROD, N, MPEROD, M, A, B, C, IDIMF,
F, IERROR)
С
С
     COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION
IS
С
C
     U(X,Y) = Y^{**}4^{*}SIN(X)
C
     ERR = 0.
     DO I = 1, M
        DO J = 1, N
           T = ABS(F(I,J)-Y(J)**4*SIN(X(I)))
           ERR = AMAX1 (T, ERR)
        END DO
     END DO
     WRITE (*, *) ' POISTG TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 5.6417E-4'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0, Discretization
Error = 5.6183E-4'
     WRITE (*, *) ' The output from your computer
is: '
     WRITE (*, *) ' IERROR =', IERROR, '
```

```
Discretization Error = ',

1 ERR

STOP

END PROGRAM TPOISTG
```

TSEPELI

```
С
     file tsepeli.f
С
С
С
С
                         copyright (c) 2005 by UCAR
С
              University Corporation for Atmospheric
Research
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
С
                        A Package of Fortran 77 and 90
С
С
*
С
                       Subroutines and Example Programs
*
С
```

```
for Modeling Geophysical Processes
С
С
                                   by
С
С
              John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                   of
С
               the National Center for Atmospheric
Research
С
                      Boulder, Colorado (80307)
U.S.A.
C
С
                         which is sponsored by
С
                    the National Science Foundation
C
С
     PROGRAM TSEPELI
     USE fish
     implicit none
     TYPE (fishworkspace) :: w
C Local Variables
INTEGER::M,N,NX,NY,I,J,MBDCND,NBDCND,IDMN,INTL,IORDER,IE
```

```
RROR
     REAL , DIMENSION (33, 33) :: USOL, GRHS
     REAL , DIMENSION (33) :: BDA, BDB
     REAL :: A, B, C, D, DLX, DLY, X, AF, BF, CF, Y,
DF, EF, FF, ALPHA
     1 , BETA, DUM, PERTRB, ERR, ERR2, ERR4
C
     DECLARE COEFFICIENT SUBROUTINES EXTERNAL
С
     external cofx, cofy
С
С
     DEFINE ARITHMETIC FUNCTIONS GIVING EXACT SOLUTION
С
С
С
     SET LIMITS ON REGION
С
     A = 0.0
     B = 1.0
     C = 0.0
     D = 1.0
C
С
     SET GRID SIZE
С
     M = 32
     N = 32
     DLX = (B - A)/FLOAT(M)
      DLY = (D - C)/FLOAT(N)
     NX = M + 1
     NY = N + 1
     DO I = 1, NX
        X = A + FLOAT(I - 1)*DLX
С
C
      SET SPECIFIED BOUNDARY CONDITIONS AT Y=C,D
С
         USOL(I,1) = UE(X,C)
         USOL(I,NY) = UE(X,D)
         CALL COFX (X, AF, BF, CF)
         DO J = 1, NY
            Y = C + FLOAT(J - 1)*DLY
            CALL COFY (Y, DF, EF, FF)
C
С
     SET RIGHT HAND SIDE
```

```
GRHS(I,J) = AF*UXXE(X,Y) + BF*UXE(X,Y) +
CF*UE(X,Y) + DF*
              UYYE(X,Y) + EF*UYE(X,Y) + FF*UE(X,Y)
         END DO
      END DO
C
С
      SET MIXED BOUNDARY CONDITIONS AT X=A, B
С
      ALPHA = 1.0
      BETA = 1.0
      DO J = 1, NY
        Y = C + FLOAT(J - 1)*DLY
         BDA(J) = UXE(A, Y) + ALPHA*UE(A, Y)
         BDB(J) = UXE(B,Y) + BETA*UE(B,Y)
      END DO
С
С
      SET BOUNDARY SWITHCES
С
      MBDCND = 3
      NBDCND = 1
С
C
      SET FIRST DIMENSION OF USOL, GRHS
C
      IDMN = 33
      set for initialization of sepeli
      INTL = 0
C
С
      OBTAIN SECOND ORDER APPROXIMATION
С
      IORDER = 2
      CALL SEPELI (INTL, IORDER, A, B, M, MBDCND, BDA,
ALPHA, BDB, BETA
     1 , C, D, N, NBDCND, DUM, DUM, DUM, DUM, COFX,
COFY, GRHS, USOL,
     2 IDMN, W, PERTRB, IERROR)
      ERR = 0.0
      DO I = 1, NX
         X = A + FLOAT(I - 1)*DLX
         DO J = 1, NY
            Y = C + FLOAT(J - 1)*DLY
            ERR = AMAX1 (ERR, ABS ((USOL(I, J) -
UE(X,Y))/UE(X,Y))
         END DO
      END DO
```

```
ERR2 = ERR
С
C
     OBTAIN FOURTH ORDER APPROXIMATION
C
     IORDER = 4
С
С
     NON-INITIAL CALL
C
     INTL = 1
     CALL SEPELI (INTL, IORDER, A, B, M, MBDCND, BDA,
ALPHA, BDB, BETA
    1 , C, D, N, NBDCND, DUM, DUM, DUM, DUM, COFX,
COFY, GRHS, USOL,
     2 IDMN, W, PERTRB, IERROR)
С
С
     COMPUTE DISCRETIZATION ERROR
С
     ERR = 0.0
     DO J = 1, NY
        Y = C + FLOAT(J - 1)*DLY
        DO I = 1, NX
           X = A + FLOAT(I - 1)*DLX
           ERR = AMAX1 (ERR, ABS ((USOL(I, J) -
UE(X,Y))/UE(X,Y))
        END DO
     END DO
     ERR4 = ERR
     Print earlier output from platforms with 32 and 64
bit floating point
     arithemtic followed by the output from this
computer
     WRITE (*, *) ' SEPELI TEST RUN *** '
     WRITE (*, *)
     1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0'
     WRITE (*, *) '
                      Second Order Discretization
Error = 9.7891E-5'
      WRITE (*, *) ' Fourth Order Discretization
Error = 1.4735E-6'
     WRITE (*, *)
     1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0'
```

```
WRITE (*, *) ' Second Order Discretization
Error = 1.2708E-4
     WRITE (*, *) ' Fourth Order Discretization
Error = 3.1948E-5'
                      The output from your computer
     WRITE (*, *) '
is: '
     WRITE (*, *) ' IERROR =', IERROR
     WRITE (*, *) ' Second Order Discretization
Error =', ERR2
     WRITE (*, *) ' Fourth Order Discretization
Error =', ERR4
     release dynamically allocated real and complex
work space
     CALL FISHFIN (W)
     STOP
     CONTAINS
     REAL FUNCTION UE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UE = (S*T)**3 + 1.0
     RETURN
     END FUNCTION UE
     REAL FUNCTION UXE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UXE = 3.0*S**2*T**3
     RETURN
     END FUNCTION UXE
     REAL FUNCTION UXXE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UXXE = 6.0*S*T**3
     RETURN
     END FUNCTION UXXE
     REAL FUNCTION UYE (S, T)
     REAL, INTENT(IN) :: S
```

```
REAL, INTENT(IN) :: T
     UYE = 3.0*S**3*T**2
     RETURN
     END FUNCTION UYE
     REAL FUNCTION UYYE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UYYE = 6.0*S**3*T
     RETURN
     END FUNCTION UYYE
     END PROGRAM TSEPELI
     SUBROUTINE COFX (X, AF, BF, CF)
C-----
C Dummy Arguments
C-----
     REAL , INTENT(IN) :: X
     REAL , INTENT (OUT) :: AF
     REAL , INTENT (OUT) :: BF
     REAL , INTENT (OUT) :: CF
С
     SET COEFFICIENTS IN THE X-DIRECTION.
С
С
     AF = (X + 1.) **2
     BF = 2.0*(X + 1.)
     CF = -X
     RETURN
     END SUBROUTINE COFX
     SUBROUTINE COFY (Y, DF, EF, FF)
  Dummy Arguments
     REAL , INTENT(IN) :: Y
     REAL , INTENT (OUT) :: DF
     REAL , INTENT (OUT) :: EF
    REAL , INTENT (OUT) :: FF
```

```
C SET COEFFICIENTS IN Y DIRECTION
C

DF = EXP(Y)
EF = 0.0
FF = -Y
RETURN
END SUBROUTINE COFY
```

TSEPX4

```
C
      file tsepx4.f
С
С
                         copyright (c) 2005 by UCAR
С
С
              University Corporation for Atmospheric
Research
С
                             all rights reserved
С
С
                           FISHPACK90 version 1.1
С
С
С
                        A Package of Fortran 77 and 90
C
*
С
                       Subroutines and Example Programs
```

```
С
                     for Modeling Geophysical Processes
С
С
С
                                  by
              John Adams, Paul Swarztrauber and Roland
Sweet
С
С
                                  of
С
               the National Center for Atmospheric
Research
                      Boulder, Colorado (80307)
U.S.A.
С
                        which is sponsored by
С
С
С
                    the National Science Foundation
С
     PROGRAM TSEPX4
     implicit none
  Local Variables
```

```
INTEGER::M,N,NX,NY,I,J,MBDCND,NBDCND,IDMN,IORDER,IERROR
      REAL , DIMENSION (33, 33) :: USOL, GRHS
      REAL , DIMENSION (33) :: BDA, BDB
     REAL :: A, B, C, D, DLX, DLY, X, AF, BF, CF, Y,
ALPHA, BETA, DUM,
     1 PERTRB, ERR, ERR2, ERR4
      EXTERNAL COFX4
C
С
      DEFINE ARITHMETIC FUNCTIONS GIVING EXACT SOLUTION
С
C
С
      SET LIMITS ON REGION
С
      A = 0.0
      B = 1.0
      C = 0.0
      D = 1.0
C
      SET GRID SIZE
С
С
     M = 32
      N = 32
      DLX = (B - A)/FLOAT(M)
      DLY = (D - C)/FLOAT(N)
      NX = M + 1
      NY = N + 1
      DO I = 1, NX
         X = A + FLOAT(I - 1)*DLX
С
С
      SET SPECIFIED BOUNDARY CONDITIONS AT Y=C,D
C
         USOL(I,1) = UE(X,C)
         USOL(I,NY) = UE(X,D)
         CALL COFX4 (X, AF, BF, CF)
         DO J = 1, NY
            Y = C + FLOAT(J - 1)*DLY
С
С
      SET RIGHT HAND SIDE
С
GRHS (I,J) = AF*UXXE(X,Y) + BF*UXE(X,Y) + CF*UE(X,Y) + UYYE(X,Y)
         END DO
      END DO
```

```
С
      SET MIXED BOUNDARY CONDITIONS AT X=A, B
C
      ALPHA = 1.0
      BETA = 1.0
      DO J = 1, NY
         Y = C + FLOAT(J - 1)*DLY
         BDA(J) = UXE(A, Y) + ALPHA*UE(A, Y)
         BDB(J) = UXE(B,Y) + BETA*UE(B,Y)
      END DO
С
С
      SET BOUNDARY SWITHCES
C
      MBDCND = 3
      NBDCND = 1
С
      SET FIRST DIMENSION OF USOL, GRHS AND WORK SPACE
LENGTH
C
      IDMN = 33
С
C
      OBTAIN SECOND ORDER APPROXIMATION
C
      IORDER = 2
      CALL SEPX4 (IORDER, A, B, M, MBDCND, BDA, ALPHA,
BDB, BETA, C, D,
     1 N, NBDCND, DUM, DUM, COFX4, GRHS, USOL, IDMN,
PERTRB, IERROR)
      COMPUTE SECOND ORDER DISCRETIZATION ERROR
(RELATIVE)
     ALSO RESET SPECIFIED BOUNDARIES AND RIGHT HAND
SIDE.
      ERR = 0.0
      DO I = 1, NX
         X = A + FLOAT(I - 1)*DLX
         USOL(I,1) = UE(X,C)
         USOL(I,NY) = UE(X,D)
         CALL COFX4 (X, AF, BF, CF)
         DO J = 1, NY
            Y = C + FLOAT(J - 1)*DLY
            ERR = AMAX1 (ERR, ABS ((USOL(I, J) -
UE(X,Y))/UE(X,Y))
```

```
RESET RIGHT HAND SIDE IN GRHS FOR FOURTH ORDER
С
APPROXIMATION CALL
GRHS(I,J) = AF*UXXE(X,Y) + BF*UXE(X,Y) + CF*UE(X,Y) + UYYE(X,Y)
        END DO
     END DO
     ERR2 = ERR
С
С
    OBTAIN FOURTH ORDER APPROXIMATION
C
     IORDER = 4
     CALL SEPX4 (IORDER, A, B, M, MBDCND, BDA, ALPHA,
BDB, BETA, C, D,
    1 N, NBDCND, DUM, DUM, COFX4, GRHS, USOL, IDMN,
PERTRB, IERROR)
C
    COMPUTE FOURTH ORDER DISCRETIZATION ERROR
(RELATIVE)
C
     ERR = 0.0
     DO J = 1, NY
        Y = C + FLOAT(J - 1)*DLY
        DO I = 1, NX
           X = A + FLOAT(I - 1)*DLX
           ERR = AMAX1 (ERR, ABS ((USOL(I, J) -
UE(X,Y))/UE(X,Y))
        END DO
     END DO
     ERR4 = ERR
     WRITE (*, *) ' SEPEX4 TEST RUN *** '
     WRITE (*, *)
    1 ' Previous 64 bit floating point arithmetic
result '
     WRITE (*, *) '
Error = 1.5985E-4'
     WRITE (*, *) ' Fourth Order Discretization
Error = 1.8575E-6'
     WRITE (*, *)
    1 ' Previous 32 bit floating point arithmetic
result '
     WRITE (*, *) ' IERROR = 0'
```

```
WRITE (*, *) ' Second Order Discretization
Error = 1.5044E-4
     WRITE (*, *) ' Fourth Order Discretization
Error = 1.5736E-5'
                       The output from your computer
     WRITE (*, *) '
is: '
     WRITE (*, *) ' IERROR =', IERROR
     WRITE (*, *) ' Second Order Discretization
Error =', ERR2
     WRITE (*, *) ' Fourth Order Discretization
Error =', ERR4
     STOP
     CONTAINS
     REAL FUNCTION UE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UE = (S*T)**3 + 1.0
     RETURN
     END FUNCTION UE
     REAL FUNCTION UXE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UXE = 3.0*S**2*T**3
     RETURN
     END FUNCTION UXE
     REAL FUNCTION UXXE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UXXE = 6.0*S*T**3
     RETURN
      END FUNCTION UXXE
     REAL FUNCTION UYE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UYE = 3.0*S**3*T**2
      RETURN
```

```
END FUNCTION UYE
     REAL FUNCTION UYYE (S, T)
     REAL, INTENT(IN) :: S
     REAL, INTENT(IN) :: T
     UYYE = 6.0*S**3*T
     RETURN
     END FUNCTION UYYE
     END PROGRAM TSEPX4
     SUBROUTINE COFX4 (X, AF, BF, CF)
C-----
  Dummy Arguments
     REAL , INTENT(IN) :: X
     REAL , INTENT (OUT) :: AF
     REAL , INTENT (OUT) :: BF
     REAL , INTENT (OUT) :: CF
С
     SET COEFFICIENTS IN THE X-DIRECTION.
С
С
     AF = (X + 1.) **2
     BF = 2.0*(X + 1.)
     CF = -X
     RETURN
     END SUBROUTINE COFX4
```

FISHPACK90, Copyright (C) 2004-2011, Computational Information Systems Laboratory, University Corporation for Atmospheric Research