**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 [details of the changes between FISHPACK and FISHPACK90.](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#details)

This [older FISHPACK document](http://opensky.library.ucar.edu/collections/TECH-NOTE-000-000-000-140) 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) | [hwscrt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscrt.txt) | [thwscrt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscrt.txt) |
| 2D Helmholtz in polar coordinates (centered grid) | [hwsplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwsplr.txt) | [thwsplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwsplr.txt) |
| 2D Helmholtz in cylindrical coordinates (centered grid) | [hwscyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscyl.txt) | [thwscyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscyl.txt) |
| 2D Helmholtz in spherical coordinates (centered grid) | [hwsssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwsssp.txt) | [thwsssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwsssp.txt) |
| 2D Helmholtz in spherical coordinates (centered grid, axisymmetric) | [hwscsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscsp.txt) | [thwscsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscsp.txt) |
| 2D Helmholtz in Cartesian coordinates (staggered grid) | [hstcrt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcrt.txt) | [thstcrt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcrt.txt) |
| 2D Helmholtz in polar coordinates (staggered grid) | [hstplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstplr.txt) | [thstplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstplr.txt) |
| 2D Helmholtz in cylindrical coordinates (staggered grid) | [hstcyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcyl.txt) | [thstcyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcyl.txt) |
| 2D Helmholtz in spherical coordinates (staggered grid) | [hstssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstssp.txt) | [thstssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstssp.txt) |
| 2D Helmholtz in spherical coordinates (staggered grid, axisymmetric) | [hstcsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcsp.txt) | [thstcsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcsp.txt) |
| 3D Helmholtz in Cartesian coordinates (centered grid) | [hw3crt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hw3crt.txt) | [thw3crt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thw3crt.txt) |
| 2D General Separable PDE (second or fourth order, centered grid) | [sepeli](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#sepeli.txt) | [tsepeli](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tsepeli.txt) |
| 2D Separable PDE (second or fourth order, centered grid) | [sepx4](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#sepx4.txt) | [tsepx4](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tsepx4.txt) |
| real linear systems solver (centered grid, **sepx4**) | [genbun](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#genbun.txt) | [tgenbun](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tgenbun.txt) |
| real block tridiagonal linear systems solver (centered grid, **sepeli**) | [blktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#blktri.txt) | [tblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tblktri.txt) |
| real linear systems solver (staggered grid) | [poistg](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#poistg.txt) | [tpoistg](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tpoistg.txt) |
| real linear systems solver (3D, centered grid) | [pois3d](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#pois3d.txt) | [tpois3d](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tpois3d.txt) |
| complex linear systems solver (centered grid) | [cmbnbn](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#cmgnbn.txt) | [tcmgnbn](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tcmgnbn.txt) |
| complex block tridiagonal linear systems solver (centered grid) | [cblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#cblktri.txt) | [tcblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tcblktri.txt) |
| real and complex fft package | [fftpack](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#fftpack.txt) | 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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscrt.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscrt.txt)

[hwsplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwsplr.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwsplr.txt)

[hwscyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscyl.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscyl.txt)

[hwsssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwsssp.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwsssp.txt)

[hwscsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hwscsp.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thwscsp.txt)

[hstcrt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcrt.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcrt.txt)

[hstplr](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstplr.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstplr.txt)

[hstcyl](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcyl.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcyl.txt)

[hwsssp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstssp.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstssp.txt)

[hstcsp](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hstcsp.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thstcsp.txt)

[hw3crt](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#hw3crt.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#thw3crt.txt)

[sepx4](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#sepx4.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tsepx4.txt)

[sepeli](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#sepeli.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tsepeli.txt)

[genbun](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#genbun.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tgenbun.txt)

[blktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#blktri.txt)

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: [tblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tblktri.txt)

[poistg](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#poistg.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tpoistg.txt)

[pois3d](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#pois3d.txt)

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](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tpois3d.txt)

[cmgnbn](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#cmgnbn.txt)

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: [tcmgnbn](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tcmgnbn.txt)

[cblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#cblktri.txt)

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: [tcblktri](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#tcblktri.txt)

**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,hstplr.f,hstssp.f,hw3crt.f,  
hwscrt.f,hwscyl.f,hwsplr.f,hwsssp.f,pois3d.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,sepeli.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**](https://www2.cisl.ucar.edu/resources/legacy/fishpack90/documentation#beginning)

**Text Below Contains Internal Files Referenced by Above Links**

**BLKTRI**

C  
C file blktri.txt (documentation for the FISHPACK solver BLKTRI)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE BLKTRI (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDIMY,Y,  
C + IERROR,W)  
C  
C  
C   
C DIMENSION OF AN(N),BN(N),CN(N),AM(M),BM(M),CM(M),Y(IDIMY,N),  
C ARGUMENTS  
C   
C LATEST REVISION JUNE 2004  
C   
C USAGE CALL BLKTRI (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,   
C CM,IDIMY,Y,IERROR,W)   
C   
C PURPOSE BLKTRI SOLVES A SYSTEM OF LINEAR EQUATIONS   
C OF THE FORM   
C   
C AN(J)\*X(I,J-1) + AM(I)\*X(I-1,J) +   
C (BN(J)+BM(I))\*X(I,J) + CN(J)\*X(I,J+1) +   
C CM(I)\*X(I+1,J) = Y(I,J)   
C   
C FOR I = 1,2,...,M AND J = 1,2,...,N.   
C   
C I+1 AND I-1 ARE EVALUATED MODULO M AND   
C J+1 AND J-1 MODULO N, I.E.,   
C   
C X(I,0) = X(I,N), X(I,N+1) = X(I,1),   
C X(0,J) = X(M,J), X(M+1,J) = X(1,J).   
C   
C THESE EQUATIONS USUALLY RESULT FROM THE   
C DISCRETIZATION OF SEPARABLE ELLIPTIC   
C EQUATIONS. BOUNDARY CONDITIONS MAY BE   
C DIRICHLET, NEUMANN, OR PERIODIC.   
C   
C ARGUMENTS   
C   
C ON INPUT IFLG   
C   
C = 0 INITIALIZATION ONLY.   
C CERTAIN QUANTITIES THAT DEPEND ON NP,   
C N, AN, BN, AND CN ARE COMPUTED AND   
C STORED IN DERIVED data type w (see  
c 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).   
C   
C NOTE:   
C A CALL WITH IFLG=0 TAKES   
C APPROXIMATELY ONE HALF THE TIME   
C 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.   
C   
C NP   
C = 0 IF AN(1) AND CN(N) ARE NOT ZERO,   
C WHICH CORRESPONDS TO PERIODIC   
C BOUNARY CONDITIONS.   
C   
C = 1 IF AN(1) AND CN(N) ARE ZERO.   
C   
C N   
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.   
C N MUST BE GREATER THAN 4.   
C THE OPERATION COUNT IS PROPORTIONAL TO   
C MNLOG2(N), HENCE N SHOULD BE SELECTED   
C LESS THAN OR EQUAL TO M.   
C   
C AN,BN,CN   
C ONE-DIMENSIONAL ARRAYS OF LENGTH N   
C THAT SPECIFY THE COEFFICIENTS IN THE   
C LINEAR EQUATIONS GIVEN ABOVE.   
C   
C MP   
C = 0 IF AM(1) AND CM(M) ARE NOT ZERO,   
C WHICH CORRESPONDS TO PERIODIC   
C BOUNDARY CONDITIONS.   
C   
C = 1 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   
C IDIMY   
C THE ROW (OR FIRST) DIMENSION OF THE   
C TWO-DIMENSIONAL ARRAY Y AS IT APPEARS   
C IN THE PROGRAM CALLING BLKTRI.   
C THIS PARAMETER IS USED TO SPECIFY THE   
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 SYSTEM OF EQUATIONS GIVEN ABOVE.   
C Y MUST BE DIMENSIONED AT LEAST M\*N.   
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 calling BLKTRI must be:  
c  
c USE fish  
c  
c Additionally the declarative statement  
c  
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 BLKTRI. The second statement  
c declares a derived type variable (defined in  
c the module "fish.f") which is used internally  
c in BLKTRI 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  
c 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 (IFLG=0) call to BLKTRI. These  
c must be preserved on non-initial calls (IFLG=1)  
c to BLKTRI. This eliminates redundant calculations  
c and saves compute time.  
c \*\*\*\* IMPORTANT! The user program calling BLKTRI should  
c include the statement:  
c  
c CALL FISHFIN(W)  
C  
C after the final approximation is generated by  
C BLKTRI. The will deallocate the real and complex  
c work space of W. Failure to include this statement  
c could result in serious memory leakage.  
C  
C   
C ARGUMENTS   
C   
C ON OUTPUT Y   
C CONTAINS THE SOLUTION X.   
C   
C IERROR   
C AN ERROR FLAG THAT INDICATES INVALID   
C INPUT PARAMETERS. EXCEPT FOR NUMBER ZER0,   
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 BLKTRI 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.   
C   
C POSSIBLE REASONS FOR THIS CONDITION ARE   
C 1. THE ARRAYS AN AND CN ARE NOT CORRECT   
C 2. TOO LARGE A GRID SPACING WAS USED   
C 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   
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  
C   
C W  
c The derived type (fishworkspace) variable W  
c contains real and complex values that must not  
C be destroyed if BLKTRI is called again with  
C IFLG=1.  
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))+   
C ABS(CM(I))+ABS(CN(J))   
C 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  
C   
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 work  
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   
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF   
C ELLIPTIC EQUATIONS'   
C NCAR TN/IA-109, JULY, 1975, 138 PP.   
C   
C SWARZTRAUBER P. N.,A DIRECT METHOD FOR   
C THE DISCRETE SOLUTION OF SEPARABLE   
C ELLIPTIC EQUATIONS, S.I.A.M.   
C J. NUMER. ANAL.,11(1974) PP. 1136-1150.   
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CBLKTRI**

C  
C file cblktri.txt (documentation for the FISHPACK solver CBLKTRI)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE CBLKTR (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,CM,IDIMY,Y,  
C + IERROR)  
C  
C   
C DIMENSION OF AN(N),BN(N),CN(N),AM(M),BM(M),CM(M),Y(IDIMY,N)  
C ARGUMENTS  
C   
C LATEST REVISION JUNE 2004  
C   
C PURPOSE CBLKTR SOLVES A SYSTEM OF LINEAR EQUATIONS   
C OF THE FORM   
C   
C AN(J)\*X(I,J-1) + AM(I)\*X(I-1,J) +   
C (BN(J)+BM(I))\*X(I,J) + CN(J)\*X(I,J+1) +   
C CM(I)\*X(I+1,J) = Y(I,J)   
C   
C FOR I = 1,2,...,M AND J = 1,2,...,N.   
C   
C I+1 AND I-1 ARE EVALUATED MODULO M AND   
C J+1 AND J-1 MODULO N, I.E.,   
C   
C X(I,0) = X(I,N), X(I,N+1) = X(I,1),   
C X(0,J) = X(M,J), X(M+1,J) = X(1,J).   
C   
C THESE EQUATIONS USUALLY RESULT FROM THE   
C DISCRETIZATION OF SEPARABLE ELLIPTIC   
C EQUATIONS. BOUNDARY CONDITIONS MAY BE   
C DIRICHLET, NEUMANN, OR PERIODIC.   
C   
C CBLKTRI IS A COMPLEX VERSION OF PACKAGE   
C BLKTRI ON ULIB.   
C   
C USAGE CALL CBLKTR (IFLG,NP,N,AN,BN,CN,MP,M,AM,BM,   
C CM,IDIMY,Y,IERROR,W)   
C   
C ARGUMENTS   
C   
C ON INPUT IFLG   
C   
C = 0 INITIALIZATION ONLY.   
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   
C = 1 THE QUANTITIES THAT WERE COMPUTED   
C IN THE INITIALIZATION ARE USED   
C TO 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.   
C HOWEVER, THE INITIALIZATION DOES   
C NOT HAVE TO BE REPEATED UNLESS NP,   
C N, AN, BN, OR CN CHANGE.   
C   
C NP   
C = 0 IF AN(1) AND CN(N) ARE NOT ZERO,   
C WHICH CORRESPONDS TO PERIODIC   
C BOUNARY CONDITIONS.   
C   
C = 1 IF AN(1) AND CN(N) ARE ZERO.   
C   
C N   
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.   
C N MUST BE GREATER THAN 4.   
C THE OPERATION COUNT IS PROPORTIONAL TO   
C MNLOG2(N), HENCE N SHOULD BE SELECTED   
C LESS THAN OR EQUAL TO M.   
C   
C AN,BN,CN   
C ONE-DIMENSIONAL ARRAYS OF LENGTH N   
C THAT SPECIFY THE COEFFICIENTS IN THE   
C LINEAR EQUATIONS GIVEN ABOVE.   
C   
C MP   
C = 0 IF AM(1) AND CM(M) ARE NOT ZERO,   
C WHICH CORRESPONDS TO PERIODIC   
C BOUNDARY CONDITIONS.   
C   
C = 1 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 COMPLEX ONE-DIMENSIONAL ARRAYS OF LENGTH M   
C THAT SPECIFY THE COEFFICIENTS IN THE LINEAR   
C EQUATIONS GIVEN ABOVE.   
C   
C IDIMY   
C THE ROW (OR FIRST) DIMENSION OF THE   
C TWO-DIMENSIONAL ARRAY Y AS IT APPEARS   
C IN THE PROGRAM CALLING CBLKTR.   
C THIS PARAMETER IS USED TO SPECIFY THE   
C VARIABLE DIMENSION OF Y.   
C IDIMY MUST BE AT LEAST M.   
C   
C Y   
C A COMPLEX TWO-DIMENSIONAL ARRAY THAT   
C SPECIFIES THE VALUES OF THE RIGHT SIDE OF   
C THE LINEAR SYSTEM OF EQUATIONS GIVEN ABOVE.   
C Y MUST BE DIMENSIONED Y(IDIMY,N) WITH   
C IDIMY .GE. M.   
C   
C W  
c A fortran 90 derived TYPE (fishworkspace) variable  
c that must be declared by the user. The first  
c two declarative statements in the user program  
c calling CBLKTRI must be:  
c  
c USE fish  
c  
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  
c declares a derived type variable (defined in  
c the module "fish.f") which is used internally  
c in CBLKTRI 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  
c 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 (IFLG=0) call to CBLKTRI. These  
c must be preserved on non-initial calls (IFLG=1)  
c to CBLKTRI. This eliminates redundant calculations  
c and saves compute time.  
c \*\*\*\* IMPORTANT! The user program calling CBLKTRI should  
c include the statement:  
c  
c CALL FISHFIN(W)  
C  
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  
c could result in serious memory leakage.  
c  
C   
C ARGUMENTS   
C   
C ON OUTPUT Y   
C CONTAINS THE SOLUTION X.   
C   
C IERROR   
C AN ERROR FLAG THAT INDICATES INVALID   
C INPUT PARAMETERS. EXCEPT FOR NUMBER ZER0,   
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.   
C   
C POSSIBLE REASONS FOR THIS CONDITION ARE   
C 1. THE ARRAYS AN AND CN ARE NOT CORRECT   
C 2. TOO LARGE A GRID SPACING WAS USED   
C 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  
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  
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))+   
C ABS(CM(I))+ABS(CN(J))   
C 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   
C I/O NONE   
C   
C PRECISION SINGLE   
C   
C REQUIRED LIBRARY comf.f,fish.f  
C FILES  
C   
C LANGUAGE FORTRAN 90  
C   
C HISTORY WRITTEN BY PAUL SWARZTRAUBER AT NCAR IN   
C THE EARLY 1970'S. REWRITTEN AN RELEASED   
C ON NCAR'S PUBLIC SOFTWARE LIBRARIES IN   
C JANUARY, 1980. Revised in June 2004 by John  
C Adams using Fortan 90 dynamically allocated  
c space and derived data types to eliminate mixed  
c mode conflicts in the earlier versions.  
C   
C ALGORITHM GENERALIZED CYCLIC REDUCTION   
C (SEE REFERENCE BELOW)   
C   
C PORTABILITY  
C THE APPROXIMATE MACHINE ACCURACY IS COMPUTED   
C IN FUNCTION EPMACH   
C   
C REFERENCES SWARZTRAUBER,P. AND R. SWEET, 'EFFICIENT   
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF   
C ELLIPTIC EQUATIONS'   
C NCAR TN/IA-109, JULY, 1975, 138 PP.   
C   
C SWARZTRAUBER P. N.,A DIRECT METHOD FOR   
C THE DISCRETE SOLUTION OF SEPARABLE   
C ELLIPTIC EQUATIONS, S.I.A.M.   
C J. NUMER. ANAL.,11(1974) PP. 1136-1150.   
C   
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**CMGNBN**

C  
C file cmgnbn.txt (documentation for the FISHPACK file cmgnbn.f)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE CMGNBN (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR)  
C  
C  
C DIMENSION OF A(M),B(M),C(M),Y(IDIMY,N)  
C ARGUMENTS  
C  
C LATEST REVISION NOVEMBER 2004  
C  
C PURPOSE THE NAME OF THIS PACKAGE IS A MNEMONIC FOR THE  
C COMPLEX GENERALIZED BUNEMAN ALGORITHM.  
C IT SOLVES THE COMPLEX LINEAR SYSTEM OF EQUATION  
C  
C A(I)\*X(I-1,J) + B(I)\*X(I,J) + C(I)\*X(I+1,J)  
C + X(I,J-1) - 2.\*X(I,J) + X(I,J+1) = Y(I,J)  
C  
C FOR I = 1,2,...,M AND J = 1,2,...,N.  
C  
C INDICES I+1 AND I-1 ARE EVALUATED MODULO M,  
C I.E., X(0,J) = X(M,J) AND X(M+1,J) = X(1,J),  
C AND X(I,0) MAY EQUAL 0, X(I,2), OR X(I,N),  
C AND X(I,N+1) MAY EQUAL 0, X(I,N-1), OR X(I,1)  
C DEPENDING ON AN INPUT PARAMETER.  
C  
C USAGE CALL CMGNBN (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,  
C IERROR)  
C  
C ARGUMENTS  
C  
C ON INPUT NPEROD  
C  
C INDICATES THE VALUES THAT X(I,0) AND  
C X(I,N+1) ARE ASSUMED TO HAVE.  
C  
C = 0 IF X(I,0) = X(I,N) AND X(I,N+1) =  
C X(I,1).  
C = 1 IF X(I,0) = X(I,N+1) = 0 .  
C = 2 IF X(I,0) = 0 AND X(I,N+1) = X(I,N-1).  
C = 3 IF X(I,0) = X(I,2) AND X(I,N+1) =  
C X(I,N-1).  
C = 4 IF X(I,0) = X(I,2) AND X(I,N+1) = 0.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.  
C N MUST BE GREATER THAN 2.  
C  
C MPEROD  
C = 0 IF A(1) AND C(M) ARE NOT ZERO  
C = 1 IF A(1) = C(M) = 0  
C  
C M  
C THE NUMBER OF UNKNOWNS IN THE I-DIRECTION.  
C N MUST BE GREATER THAN 2.  
C  
C A,B,C  
C ONE-DIMENSIONAL COMPLEX ARRAYS OF LENGTH M  
C THAT SPECIFY THE COEFFICIENTS IN THE LINEAR  
C EQUATIONS GIVEN ABOVE. IF MPEROD = 0  
C 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  
C A(I) = C(1)  
C C(I) = C(1)  
C B(I) = B(1)  
C  
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 CMGNBN.  
C THIS PARAMETER IS USED TO SPECIFY THE  
C 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  
C  
C CONTAINS THE SOLUTION X.  
C  
C IERROR  
C AN ERROR FLAG WHICH INDICATES INVALID  
C INPUT PARAMETERS EXCEPT FOR NUMBER  
C ZERO, A SOLUTION IS NOT ATTEMPTED.  
C  
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(1) OR C(I) .NE. C(1) OR  
C B(I) .NE. B(1) FOR  
C SOME I=1,2,...,M.  
C = 7 A(1) .NE. 0 OR C(M) .NE. 0 AND  
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 SPECIAL CONDITONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED LIBRARY comf.f,fish.f  
C FILES  
C  
C LANGUAGE FORTRAN 90  
C  
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.  
c Revised in June 2004 by John Adams using  
c 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  
C AND, WHEN MPEROD = 1  
C  
C A(1) = C(M) = 0  
C A(M) = C(1) = 2.  
C  
C THE SOLUTION X WAS SUBSTITUTED INTO THE  
C GIVEN SYSTEM AND A RIGHT SIDE Y WAS  
C COMPUTED. USING THIS ARRAY Y, SUBROUTINE  
C CMGNBN WAS CALLED TO PRODUCE APPROXIMATE  
C SOLUTION Z. THEN RELATIVE ERROR  
C E = MAX(CABS(Z(I,J)-X(I,J)))/  
C MAX(CABS(X(I,J)))  
C WAS COMPUTED, WHERE THE TWO MAXIMA ARE TAKEN  
C OVER I=1,2,...,M AND J=1,...,N.  
C  
C THE VALUE OF E IS GIVEN IN THE TABLE  
C BELOW FOR SOME TYPICAL VALUES OF M AND N.  
C  
C M (=N) MPEROD NPEROD E  
C ------ ------ ------ ------  
C  
C 31 0 0 1.E-12  
C 31 1 1 4.E-13  
C 31 1 3 2.E-12  
C 32 0 0 7.E-14  
C 32 1 1 5.E-13  
C 32 1 3 2.E-13  
C 33 0 0 6.E-13  
C 33 1 1 5.E-13  
C 33 1 3 3.E-12  
C 63 0 0 5.E-12  
C 63 1 1 6.E-13  
C 63 1 3 1.E-11  
C 64 0 0 1.E-13  
C 64 1 1 3.E-12  
C 64 1 3 3.E-13  
C 65 0 0 2.E-12  
C 65 1 1 5.E-13  
C 65 1 3 1.E-11  
C  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**COMF**

C  
C file comf.f  
C  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C  
C PACKAGE COMF THE ENTRIES IN THIS PACKAGE ARE LOWLEVEL  
C ENTRIES, SUPPORTING FISHPACK ENTRIES BLKTRI  
C AND CBLKTRI. THAT IS, THESE ROUTINES ARE  
C NOT CALLED DIRECTLY BY USERS, BUT RATHER  
C BY ENTRIES WITHIN BLKTRI AND CBLKTRI.  
C DESCRIPTION OF ENTRIES EPMACH AND PIMACH  
C FOLLOW BELOW.  
C  
C LATEST REVISION JUNE 2004  
C  
C SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED LIBRARY NONE  
C FILES  
C  
C LANGUAGE FORTRAN 90  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C FUNCTION EPMACH (DUM)  
C  
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  
C BLKTRI PACKAGE WAS CONVERTED FROM THE  
C CDC 7600 TO RUN ON THE CRAY-1, CALCULATED  
C MACHINE ACCURACY BY SUCCESSIVE DIVISIONS  
C BY 10. USE OF THIS CONSTANT CAUSED BLKTRI  
C TO COMPUTE SOLUTIONS ON THE CRAY-1 WITH FOUR  
C FEWER PLACES OF ACCURACY THAN THE VERSION  
C ON THE 7600. IT WAS FOUND THAT COMPUTING  
C MACHINE ACCURACY BY SUCCESSIVE DIVISIONS  
C OF 2 PRODUCED A MACHINE ACCURACY 29% LESS  
C THAN THE VALUE GENERATED BY SUCCESSIVE  
C DIVISIONS BY 10, AND THAT USE OF THIS  
C MACHINE CONSTANT IN THE BLKTRI PACKAGE  
C 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  
C THAN THE CRAY1, BUT THE RETURNED VALUE MAY  
C BE UNSATISFACTORY. SEE HISTORY ABOVE.  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C FUNCTION PIMACH (DUM)  
C  
C PURPOSE TO SUPPLY THE VALUE OF THE CONSTANT PI  
C CORRECT TO MACHINE PRECISION WHERE  
C PI=3.141592653589793238462643383279502884197  
C 1693993751058209749446  
C  
C USAGE PI = PIMACH (DUM)  
C  
C ARGUMENTS  
C ON INPUT DUM  
C DUMMY VALUE  
C  
C ARGUMENTS  
C ON OUTPUT NONE  
C  
C ALGORITHM THE VALUE OF PI IS SET TO 4.\*ATAN(1.0)  
C  
C PORTABILITY THIS ENTRY IS PORTABLE, BUT USERS SHOULD  
C CHECK TO SEE WHETHER GREATER ACCURACY IS  
C REQUIRED.  
C  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C REVISION HISTORY---  
C  
C SEPTEMBER 1973 VERSION 1  
C APRIL 1976 VERSION 2  
C JANUARY 1978 VERSION 3  
C DECEMBER 1979 VERSION 3.1  
C FEBRUARY 1985 DOCUMENTATION UPGRADE  
C NOVEMBER 1988 VERSION 3.2, FORTRAN 77 CHANGES  
C June 2004 Version 5.0, Fortran 90 changes  
C-----------------------------------------------------------------------

**FFTPACK**

C  
C file fftpack.txt (documentation for fftpack.f on fishpack)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C LATEST REVISION  
C ---------------  
C June 2004 (VERSION 5.0) FORTRAN 90 CHANGES  
C  
C PURPOSE  
C -------  
C THIS PACKAGE CONSISTS OF PROGRAMS WHICH PERFORM FAST FOURIER  
C TRANSFORMS FOR BOTH COMPLEX AND REAL PERIODIC SEQUENCES AND  
C CERTAIN OTHER SYMMETRIC SEQUENCES THAT ARE LISTED BELOW.  
C  
C USAGE  
C -----  
C 1. RFFTI INITIALIZE RFFTF AND RFFTB  
C 2. RFFTF FORWARD TRANSFORM OF A REAL PERIODIC SEQUENCE  
C 3. RFFTB BACKWARD TRANSFORM OF A REAL COEFFICIENT ARRAY  
C  
C 4. EZFFTI INITIALIZE EZFFTF AND EZFFTB  
C 5. EZFFTF A SIMPLIFIED REAL PERIODIC FORWARD TRANSFORM  
C 6. EZFFTB A SIMPLIFIED REAL PERIODIC BACKWARD TRANSFORM  
C  
C 7. SINTI INITIALIZE SINT  
C 8. SINT SINE TRANSFORM OF A REAL ODD SEQUENCE  
C  
C 9. COSTI INITIALIZE COST  
C 10. COST COSINE TRANSFORM OF A REAL EVEN SEQUENCE  
C  
C 11. SINQI INITIALIZE SINQF AND SINQB  
C 12. SINQF FORWARD SINE TRANSFORM WITH ODD WAVE NUMBERS  
C 13. SINQB UNNORMALIZED INVERSE OF SINQF  
C  
C 14. COSQI INITIALIZE COSQF AND COSQB  
C 15. COSQF FORWARD COSINE TRANSFORM WITH ODD WAVE NUMBERS  
C 16. COSQB UNNORMALIZED INVERSE OF COSQF  
C  
C 17. CFFTI INITIALIZE CFFTF AND CFFTB  
C 18. CFFTF FORWARD TRANSFORM OF A COMPLEX PERIODIC SEQUENCE  
C 19. CFFTB UNNORMALIZED INVERSE OF CFFTF  
C  
C SPECIAL CONDITIONS  
C ------------------  
C BEFORE CALLING ROUTINES EZFFTB AND EZFFTF FOR THE FIRST TIME,  
C OR BEFORE CALLING EZFFTB AND EZFFTF WITH A DIFFERENT LENGTH,  
C USERS MUST INITIALIZE BY CALLING ROUTINE EZFFTI.  
C  
C I/O  
C ---  
C NONE  
C  
C PRECISION  
C ---------  
C NONE  
C  
C REQUIRED LIBRARY FILES  
C ----------------------  
C NONE  
C  
C LANGUAGE  
C --------  
C FORTRAN 90  
C  
C HISTORY  
C -------  
C DEVELOPED AT NCAR IN BOULDER, COLORADO BY PAUL N. SWARZTRAUBER  
C OF THE SCIENTIFIC COMPUTING DIVISION. RELEASED ON NCAR'S PUBLIC  
C SOFTWARE LIBRARIES IN JANUARY 1980. MODIFIED MAY 29, 1985 TO  
C INCREASE EFFICIENCY. Fortran 90 changes made June 2004  
C  
C PORTABILITY  
C -----------  
C FORTRAN 90  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE RFFTI(N,WSAVE)  
C  
C SUBROUTINE RFFTI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C BOTH RFFTF AND RFFTB. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 2\*N+15.  
C THE SAME WORK ARRAY CAN BE USED FOR BOTH RFFTF AND RFFTB  
C AS LONG AS N REMAINS UNCHANGED. DIFFERENT WSAVE ARRAYS  
C ARE REQUIRED FOR DIFFERENT VALUES OF N. THE CONTENTS OF  
C WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF RFFTF OR RFFTB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE RFFTF(N,R,WSAVE)  
C  
C SUBROUTINE RFFTF COMPUTES THE FOURIER COEFFICIENTS OF A REAL  
C PERODIC SEQUENCE (FOURIER ANALYSIS). THE TRANSFORM IS DEFINED  
C BELOW AT OUTPUT PARAMETER R.  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY R TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C N MAY CHANGE SO LONG AS DIFFERENT WORK ARRAYS ARE PROVIDED  
C  
C R A REAL ARRAY OF LENGTH N WHICH CONTAINS THE SEQUENCE  
C TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 2\*N+15.  
C IN THE PROGRAM THAT CALLS RFFTF. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE RFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY RFFTF AND RFFTB.  
C  
C  
C OUTPUT PARAMETERS  
C  
C R R(1) = THE SUM FROM I=1 TO I=N OF R(I)  
C  
C IF N IS EVEN SET L =N/2 , IF N IS ODD SET L = (N+1)/2  
C  
C THEN FOR K = 2,...,L  
C  
C R(2\*K-2) = THE SUM FROM I = 1 TO I = N OF  
C  
C R(I)\*COS((K-1)\*(I-1)\*2\*PI/N)  
C  
C R(2\*K-1) = THE SUM FROM I = 1 TO I = N OF  
C  
C -R(I)\*SIN((K-1)\*(I-1)\*2\*PI/N)  
C  
C IF N IS EVEN  
C  
C R(N) = THE SUM FROM I = 1 TO I = N OF  
C  
C (-1)\*\*(I-1)\*R(I)  
C  
C \*\*\*\*\* NOTE  
C THIS TRANSFORM IS UNNORMALIZED SINCE A CALL OF RFFTF  
C FOLLOWED BY A CALL OF RFFTB WILL MULTIPLY THE INPUT  
C SEQUENCE BY N.  
C  
C WSAVE CONTAINS RESULTS WHICH MUST NOT BE DESTROYED BETWEEN  
C CALLS OF RFFTF OR RFFTB.  
C  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE RFFTB(N,R,WSAVE)  
C  
C SUBROUTINE RFFTB COMPUTES THE REAL PERODIC SEQUENCE FROM ITS  
C FOURIER COEFFICIENTS (FOURIER SYNTHESIS). THE TRANSFORM IS DEFINED  
C BELOW AT OUTPUT PARAMETER R.  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY R TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C N MAY CHANGE SO LONG AS DIFFERENT WORK ARRAYS ARE PROVIDED  
C  
C R A REAL ARRAY OF LENGTH N WHICH CONTAINS THE SEQUENCE  
C TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 2\*N+15.  
C IN THE PROGRAM THAT CALLS RFFTB. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE RFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY RFFTF AND RFFTB.  
C  
C  
C OUTPUT PARAMETERS  
C  
C R FOR N EVEN AND FOR I = 1,...,N  
C  
C R(I) = R(1)+(-1)\*\*(I-1)\*R(N)  
C  
C PLUS THE SUM FROM K=2 TO K=N/2 OF  
C  
C 2.\*R(2\*K-2)\*COS((K-1)\*(I-1)\*2\*PI/N)  
C  
C -2.\*R(2\*K-1)\*SIN((K-1)\*(I-1)\*2\*PI/N)  
C  
C FOR N ODD AND FOR I = 1,...,N  
C  
C R(I) = R(1) PLUS THE SUM FROM K=2 TO K=(N+1)/2 OF  
C  
C 2.\*R(2\*K-2)\*COS((K-1)\*(I-1)\*2\*PI/N)  
C  
C -2.\*R(2\*K-1)\*SIN((K-1)\*(I-1)\*2\*PI/N)  
C  
C \*\*\*\*\* NOTE  
C THIS TRANSFORM IS UNNORMALIZED SINCE A CALL OF RFFTF  
C FOLLOWED BY A CALL OF RFFTB WILL MULTIPLY THE INPUT  
C SEQUENCE BY N.  
C  
C WSAVE CONTAINS RESULTS WHICH MUST NOT BE DESTROYED BETWEEN  
C CALLS OF RFFTB OR RFFTF.  
C  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE EZFFTI(N,WSAVE)  
C  
C SUBROUTINE EZFFTI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C BOTH EZFFTF AND EZFFTB. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C THE SAME WORK ARRAY CAN BE USED FOR BOTH EZFFTF AND EZFFTB  
C AS LONG AS N REMAINS UNCHANGED. DIFFERENT WSAVE ARRAYS  
C ARE REQUIRED FOR DIFFERENT VALUES OF N.  
C  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE EZFFTF(N,R,AZERO,A,B,WSAVE)  
C  
C SUBROUTINE EZFFTF COMPUTES THE FOURIER COEFFICIENTS OF A REAL  
C PERODIC SEQUENCE (FOURIER ANALYSIS). THE TRANSFORM IS DEFINED  
C BELOW AT OUTPUT PARAMETERS AZERO,A AND B. EZFFTF IS A SIMPLIFIED  
C BUT SLOWER VERSION OF RFFTF.  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY R TO BE TRANSFORMED. THE METHOD  
C IS MUST EFFICIENT WHEN N IS THE PRODUCT OF SMALL PRIMES.  
C  
C R A REAL ARRAY OF LENGTH N WHICH CONTAINS THE SEQUENCE  
C TO BE TRANSFORMED. R IS NOT DESTROYED.  
C  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C IN THE PROGRAM THAT CALLS EZFFTF. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE EZFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY EZFFTF AND EZFFTB.  
C  
C OUTPUT PARAMETERS  
C  
C AZERO THE SUM FROM I=1 TO I=N OF R(I)/N  
C  
C A,B FOR N EVEN B(N/2)=0. AND A(N/2) IS THE SUM FROM I=1 TO  
C I=N OF (-1)\*\*(I-1)\*R(I)/N  
C  
C FOR N EVEN DEFINE KMAX=N/2-1  
C FOR N ODD DEFINE KMAX=(N-1)/2  
C  
C THEN FOR K=1,...,KMAX  
C  
C A(K) EQUALS THE SUM FROM I=1 TO I=N OF  
C  
C 2./N\*R(I)\*COS(K\*(I-1)\*2\*PI/N)  
C  
C B(K) EQUALS THE SUM FROM I=1 TO I=N OF  
C  
C 2./N\*R(I)\*SIN(K\*(I-1)\*2\*PI/N)  
C  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE EZFFTB(N,R,AZERO,A,B,WSAVE)  
C  
C SUBROUTINE EZFFTB COMPUTES A REAL PERODIC SEQUENCE FROM ITS  
C FOURIER COEFFICIENTS (FOURIER SYNTHESIS). THE TRANSFORM IS  
C DEFINED BELOW AT OUTPUT PARAMETER R. EZFFTB IS A SIMPLIFIED  
C BUT SLOWER VERSION OF RFFTB.  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE OUTPUT ARRAY R. THE METHOD IS MOST  
C EFFICIENT WHEN N IS THE PRODUCT OF SMALL PRIMES.  
C  
C AZERO THE CONSTANT FOURIER COEFFICIENT  
C  
C A,B ARRAYS WHICH CONTAIN THE REMAINING FOURIER COEFFICIENTS  
C THESE ARRAYS ARE NOT DESTROYED.  
C  
C THE LENGTH OF THESE ARRAYS DEPENDS ON WHETHER N IS EVEN OR  
C ODD.  
C  
C IF N IS EVEN N/2 LOCATIONS ARE REQUIRED  
C IF N IS ODD (N-1)/2 LOCATIONS ARE REQUIRED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C IN THE PROGRAM THAT CALLS EZFFTB. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE EZFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY EZFFTF AND EZFFTB.  
C  
C  
C OUTPUT PARAMETERS  
C  
C R IF N IS EVEN DEFINE KMAX=N/2  
C IF N IS ODD DEFINE KMAX=(N-1)/2  
C  
C THEN FOR I=1,...,N  
C  
C R(I)=AZERO PLUS THE SUM FROM K=1 TO K=KMAX OF  
C  
C A(K)\*COS(K\*(I-1)\*2\*PI/N)+B(K)\*SIN(K\*(I-1)\*2\*PI/N)  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* COMPLEX NOTATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C FOR J=1,...,N  
C  
C R(J) EQUALS THE SUM FROM K=-KMAX TO K=KMAX OF  
C  
C C(K)\*EXP(I\*K\*(J-1)\*2\*PI/N)  
C  
C WHERE  
C  
C C(K) = .5\*CMPLX(A(K),-B(K)) FOR K=1,...,KMAX  
C  
C C(-K) = CONJG(C(K))  
C  
C C(0) = AZERO  
C  
C AND I=SQRT(-1)  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\* AMPLITUDE - PHASE NOTATION \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C FOR I=1,...,N  
C  
C R(I) EQUALS AZERO PLUS THE SUM FROM K=1 TO K=KMAX OF  
C  
C ALPHA(K)\*COS(K\*(I-1)\*2\*PI/N+BETA(K))  
C  
C WHERE  
C  
C ALPHA(K) = SQRT(A(K)\*A(K)+B(K)\*B(K))  
C  
C COS(BETA(K))=A(K)/ALPHA(K)  
C  
C SIN(BETA(K))=-B(K)/ALPHA(K)  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE SINTI(N,WSAVE)  
C  
C SUBROUTINE SINTI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C SUBROUTINE SINT. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N+1 IS A PRODUCT OF SMALL PRIMES.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WITH AT LEAST INT(2.5\*N+15) LOCATIONS.  
C DIFFERENT WSAVE ARRAYS ARE REQUIRED FOR DIFFERENT VALUES  
C OF N. THE CONTENTS OF WSAVE MUST NOT BE CHANGED BETWEEN  
C CALLS OF SINT.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE SINT(N,X,WSAVE)  
C  
C SUBROUTINE SINT COMPUTES THE DISCRETE FOURIER SINE TRANSFORM  
C OF AN ODD SEQUENCE X(I). THE TRANSFORM IS DEFINED BELOW AT  
C OUTPUT PARAMETER X.  
C  
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  
C X BY 2\*(N+1).  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINT MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINTI(N,WSAVE).  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N+1 IS THE PRODUCT OF SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C  
C WSAVE A WORK ARRAY WITH DIMENSION AT LEAST INT(2.5\*N+15)  
C IN THE PROGRAM THAT CALLS SINT. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I)= THE SUM FROM K=1 TO K=N  
C  
C 2\*X(K)\*SIN(K\*I\*PI/(N+1))  
C  
C A CALL OF SINT FOLLOWED BY ANOTHER CALL OF  
C SINT WILL MULTIPLY THE SEQUENCE X BY 2\*(N+1).  
C HENCE SINT IS THE UNNORMALIZED INVERSE  
C OF ITSELF.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT BE  
C DESTROYED BETWEEN CALLS OF SINT.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE COSTI(N,WSAVE)  
C  
C SUBROUTINE COSTI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C SUBROUTINE COST. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N-1 IS A PRODUCT OF SMALL PRIMES.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C DIFFERENT WSAVE ARRAYS ARE REQUIRED FOR DIFFERENT VALUES  
C OF N. THE CONTENTS OF WSAVE MUST NOT BE CHANGED BETWEEN  
C CALLS OF COST.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE COST(N,X,WSAVE)  
C  
C SUBROUTINE COST COMPUTES THE DISCRETE FOURIER COSINE TRANSFORM  
C OF AN EVEN SEQUENCE X(I). THE TRANSFORM IS DEFINED BELOW AT OUTPUT  
C PARAMETER X.  
C  
C COST IS THE UNNORMALIZED INVERSE OF ITSELF SINCE A CALL OF COST  
C FOLLOWED BY ANOTHER CALL OF COST WILL MULTIPLY THE INPUT SEQUENCE  
C X BY 2\*(N-1). THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER X  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COST MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSTI(N,WSAVE).  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE SEQUENCE X. N MUST BE GREATER THAN 1.  
C THE METHOD IS MOST EFFICIENT WHEN N-1 IS A PRODUCT OF  
C SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15  
C IN THE PROGRAM THAT CALLS COST. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I) = X(1)+(-1)\*\*(I-1)\*X(N)  
C  
C + THE SUM FROM K=2 TO K=N-1  
C  
C 2\*X(K)\*COS((K-1)\*(I-1)\*PI/(N-1))  
C  
C A CALL OF COST FOLLOWED BY ANOTHER CALL OF  
C COST WILL MULTIPLY THE SEQUENCE X BY 2\*(N-1)  
C HENCE COST IS THE UNNORMALIZED INVERSE  
C OF ITSELF.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT BE  
C DESTROYED BETWEEN CALLS OF COST.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE SINQI(N,WSAVE)  
C  
C SUBROUTINE SINQI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C BOTH SINQF AND SINQB. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C THE SAME WORK ARRAY CAN BE USED FOR BOTH SINQF AND SINQB  
C AS LONG AS N REMAINS UNCHANGED. DIFFERENT WSAVE ARRAYS  
C ARE REQUIRED FOR DIFFERENT VALUES OF N. THE CONTENTS OF  
C WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF SINQF OR SINQB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE SINQF(N,X,WSAVE)  
C  
C SUBROUTINE SINQF COMPUTES THE FAST FOURIER TRANSFORM OF QUARTER  
C WAVE DATA. THAT IS , SINQF COMPUTES THE COEFFICIENTS IN A SINE  
C SERIES REPRESENTATION WITH ONLY ODD WAVE NUMBERS. THE TRANSFORM  
C IS DEFINED BELOW AT OUTPUT PARAMETER X.  
C  
C SINQB IS THE UNNORMALIZED INVERSE OF SINQF SINCE A CALL OF SINQF  
C FOLLOWED BY A CALL OF SINQB WILL MULTIPLY THE INPUT SEQUENCE X  
C BY 4\*N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINQF MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINQI(N,WSAVE).  
C  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY X TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C IN THE PROGRAM THAT CALLS SINQF. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINQI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I) = (-1)\*\*(I-1)\*X(N)  
C  
C + THE SUM FROM K=1 TO K=N-1 OF  
C  
C 2\*X(K)\*SIN((2\*I-1)\*K\*PI/(2\*N))  
C  
C A CALL OF SINQF FOLLOWED BY A CALL OF  
C SINQB WILL MULTIPLY THE SEQUENCE X BY 4\*N.  
C THEREFORE SINQB IS THE UNNORMALIZED INVERSE  
C OF SINQF.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT  
C BE DESTROYED BETWEEN CALLS OF SINQF OR SINQB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE SINQB(N,X,WSAVE)  
C  
C SUBROUTINE SINQB COMPUTES THE FAST FOURIER TRANSFORM OF QUARTER  
C WAVE DATA. THAT IS , SINQB COMPUTES A SEQUENCE FROM ITS  
C REPRESENTATION IN TERMS OF A SINE SERIES WITH ODD WAVE NUMBERS.  
C THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER X.  
C  
C SINQF IS THE UNNORMALIZED INVERSE OF SINQB SINCE A CALL OF SINQB  
C FOLLOWED BY A CALL OF SINQF WILL MULTIPLY THE INPUT SEQUENCE X  
C BY 4\*N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE SINQB MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINQI(N,WSAVE).  
C  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY X TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C IN THE PROGRAM THAT CALLS SINQB. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE SINQI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I)= THE SUM FROM K=1 TO K=N OF  
C  
C 4\*X(K)\*SIN((2K-1)\*I\*PI/(2\*N))  
C  
C A CALL OF SINQB FOLLOWED BY A CALL OF  
C SINQF WILL MULTIPLY THE SEQUENCE X BY 4\*N.  
C THEREFORE SINQF IS THE UNNORMALIZED INVERSE  
C OF SINQB.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT  
C BE DESTROYED BETWEEN CALLS OF SINQB OR SINQF.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE COSQI(N,WSAVE)  
C  
C SUBROUTINE COSQI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C BOTH COSQF AND COSQB. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE ARRAY TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15.  
C THE SAME WORK ARRAY CAN BE USED FOR BOTH COSQF AND COSQB  
C AS LONG AS N REMAINS UNCHANGED. DIFFERENT WSAVE ARRAYS  
C ARE REQUIRED FOR DIFFERENT VALUES OF N. THE CONTENTS OF  
C WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF COSQF OR COSQB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE COSQF(N,X,WSAVE)  
C  
C SUBROUTINE COSQF COMPUTES THE FAST FOURIER TRANSFORM OF QUARTER  
C WAVE DATA. THAT IS , COSQF COMPUTES THE COEFFICIENTS IN A COSINE  
C SERIES REPRESENTATION WITH ONLY ODD WAVE NUMBERS. THE TRANSFORM  
C IS DEFINED BELOW AT OUTPUT PARAMETER X  
C  
C COSQF IS THE UNNORMALIZED INVERSE OF COSQB SINCE A CALL OF COSQF  
C FOLLOWED BY A CALL OF COSQB WILL MULTIPLY THE INPUT SEQUENCE X  
C BY 4\*N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COSQF MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSQI(N,WSAVE).  
C  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY X TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 3\*N+15  
C IN THE PROGRAM THAT CALLS COSQF. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSQI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I) = X(1) PLUS THE SUM FROM K=2 TO K=N OF  
C  
C 2\*X(K)\*COS((2\*I-1)\*(K-1)\*PI/(2\*N))  
C  
C A CALL OF COSQF FOLLOWED BY A CALL OF  
C COSQB WILL MULTIPLY THE SEQUENCE X BY 4\*N.  
C THEREFORE COSQB IS THE UNNORMALIZED INVERSE  
C OF COSQF.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT  
C BE DESTROYED BETWEEN CALLS OF COSQF OR COSQB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE COSQB(N,X,WSAVE)  
C  
C SUBROUTINE COSQB COMPUTES THE FAST FOURIER TRANSFORM OF QUARTER  
C WAVE DATA. THAT IS , COSQB COMPUTES A SEQUENCE FROM ITS  
C REPRESENTATION IN TERMS OF A COSINE SERIES WITH ODD WAVE NUMBERS.  
C THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER X.  
C  
C COSQB IS THE UNNORMALIZED INVERSE OF COSQF SINCE A CALL OF COSQB  
C FOLLOWED BY A CALL OF COSQF WILL MULTIPLY THE INPUT SEQUENCE X  
C BY 4\*N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE COSQB MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSQI(N,WSAVE).  
C  
C  
C INPUT PARAMETERS  
C  
C N THE LENGTH OF THE ARRAY X TO BE TRANSFORMED. THE METHOD  
C IS MOST EFFICIENT WHEN N IS A PRODUCT OF SMALL PRIMES.  
C  
C X AN ARRAY WHICH CONTAINS THE SEQUENCE TO BE TRANSFORMED  
C  
C WSAVE A WORK ARRAY THAT MUST BE DIMENSIONED AT LEAST 3\*N+15  
C IN THE PROGRAM THAT CALLS COSQB. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE COSQI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C  
C OUTPUT PARAMETERS  
C  
C X FOR I=1,...,N  
C  
C X(I)= THE SUM FROM K=1 TO K=N OF  
C  
C 4\*X(K)\*COS((2\*K-1)\*(I-1)\*PI/(2\*N))  
C  
C A CALL OF COSQB FOLLOWED BY A CALL OF  
C COSQF WILL MULTIPLY THE SEQUENCE X BY 4\*N.  
C THEREFORE COSQF IS THE UNNORMALIZED INVERSE  
C OF COSQB.  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT  
C BE DESTROYED BETWEEN CALLS OF COSQB OR COSQF.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE CFFTI(N,WSAVE)  
C  
C SUBROUTINE CFFTI INITIALIZES THE ARRAY WSAVE WHICH IS USED IN  
C BOTH CFFTF AND CFFTB. THE PRIME FACTORIZATION OF N TOGETHER WITH  
C A TABULATION OF THE TRIGONOMETRIC FUNCTIONS ARE COMPUTED AND  
C STORED IN WSAVE.  
C  
C INPUT PARAMETER  
C  
C N THE LENGTH OF THE SEQUENCE TO BE TRANSFORMED  
C  
C OUTPUT PARAMETER  
C  
C WSAVE A WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 4\*N+15  
C THE SAME WORK ARRAY CAN BE USED FOR BOTH CFFTF AND CFFTB  
C AS LONG AS N REMAINS UNCHANGED. DIFFERENT WSAVE ARRAYS  
C ARE REQUIRED FOR DIFFERENT VALUES OF N. THE CONTENTS OF  
C WSAVE MUST NOT BE CHANGED BETWEEN CALLS OF CFFTF OR CFFTB.  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE CFFTF(N,C,WSAVE)  
C  
C SUBROUTINE CFFTF COMPUTES THE FORWARD COMPLEX DISCRETE FOURIER  
C TRANSFORM (THE FOURIER ANALYSIS). EQUIVALENTLY , CFFTF COMPUTES  
C THE FOURIER COEFFICIENTS OF A COMPLEX PERIODIC SEQUENCE.  
C THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER C.  
C  
C THE TRANSFORM IS NOT NORMALIZED. TO OBTAIN A NORMALIZED TRANSFORM  
C THE OUTPUT MUST BE DIVIDED BY N. OTHERWISE A CALL OF CFFTF  
C FOLLOWED BY A CALL OF CFFTB WILL MULTIPLY THE SEQUENCE BY N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE CFFTF MUST BE  
C INITIALIZED BY CALLING SUBROUTINE CFFTI(N,WSAVE).  
C  
C INPUT PARAMETERS  
C  
C  
C N THE LENGTH OF THE COMPLEX SEQUENCE C. THE METHOD IS  
C MORE EFFICIENT WHEN N IS THE PRODUCT OF SMALL PRIMES. N  
C  
C C A COMPLEX ARRAY OF LENGTH N WHICH CONTAINS THE SEQUENCE  
C  
C WSAVE A REAL WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 4N+15  
C IN THE PROGRAM THAT CALLS CFFTF. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE CFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY CFFTF AND CFFTB.  
C  
C OUTPUT PARAMETERS  
C  
C C FOR J=1,...,N  
C  
C C(J)=THE SUM FROM K=1,...,N OF  
C  
C C(K)\*EXP(-I\*(J-1)\*(K-1)\*2\*PI/N)  
C  
C WHERE I=SQRT(-1)  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT BE  
C DESTROYED BETWEEN CALLS OF SUBROUTINE CFFTF OR CFFTB  
C  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
C  
C SUBROUTINE CFFTB(N,C,WSAVE)  
C  
C SUBROUTINE CFFTB COMPUTES THE BACKWARD COMPLEX DISCRETE FOURIER  
C TRANSFORM (THE FOURIER SYNTHESIS). EQUIVALENTLY , CFFTB COMPUTES  
C A COMPLEX PERIODIC SEQUENCE FROM ITS FOURIER COEFFICIENTS.  
C THE TRANSFORM IS DEFINED BELOW AT OUTPUT PARAMETER C.  
C  
C A CALL OF CFFTF FOLLOWED BY A CALL OF CFFTB WILL MULTIPLY THE  
C SEQUENCE BY N.  
C  
C THE ARRAY WSAVE WHICH IS USED BY SUBROUTINE CFFTB MUST BE  
C INITIALIZED BY CALLING SUBROUTINE CFFTI(N,WSAVE).  
C  
C INPUT PARAMETERS  
C  
C  
C N THE LENGTH OF THE COMPLEX SEQUENCE C. THE METHOD IS  
C MORE EFFICIENT WHEN N IS THE PRODUCT OF SMALL PRIMES.  
C  
C C A COMPLEX ARRAY OF LENGTH N WHICH CONTAINS THE SEQUENCE  
C  
C WSAVE A REAL WORK ARRAY WHICH MUST BE DIMENSIONED AT LEAST 4N+15  
C IN THE PROGRAM THAT CALLS CFFTB. THE WSAVE ARRAY MUST BE  
C INITIALIZED BY CALLING SUBROUTINE CFFTI(N,WSAVE) AND A  
C DIFFERENT WSAVE ARRAY MUST BE USED FOR EACH DIFFERENT  
C VALUE OF N. THIS INITIALIZATION DOES NOT HAVE TO BE  
C REPEATED SO LONG AS N REMAINS UNCHANGED THUS SUBSEQUENT  
C TRANSFORMS CAN BE OBTAINED FASTER THAN THE FIRST.  
C THE SAME WSAVE ARRAY CAN BE USED BY CFFTF AND CFFTB.  
C  
C OUTPUT PARAMETERS  
C  
C C FOR J=1,...,N  
C  
C C(J)=THE SUM FROM K=1,...,N OF  
C  
C C(K)\*EXP(I\*(J-1)\*(K-1)\*2\*PI/N)  
C  
C WHERE I=SQRT(-1)  
C  
C WSAVE CONTAINS INITIALIZATION CALCULATIONS WHICH MUST NOT BE  
C DESTROYED BETWEEN CALLS OF SUBROUTINE CFFTF OR CFFTB  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**GENBUN**

C  
C file genbun.txt (documentation for the FISHPACK solver GENBUN)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE GENBUN (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR)  
C  
C  
C DIMENSION OF A(M),B(M),C(M),Y(IDIMY,N)  
C ARGUMENTS  
C  
C LATEST REVISION JUNE 2004  
C  
C PURPOSE THE NAME OF THIS PACKAGE IS A MNEMONIC FOR THE  
C GENERALIZED BUNEMAN ALGORITHM.  
C  
C IT SOLVES THE REAL LINEAR SYSTEM OF EQUATIONS  
C  
C A(I)\*X(I-1,J) + B(I)\*X(I,J) + C(I)\*X(I+1,J)  
C + X(I,J-1) - 2.\*X(I,J) + X(I,J+1) = Y(I,J)  
C  
C FOR I = 1,2,...,M AND J = 1,2,...,N.  
C  
C INDICES I+1 AND I-1 ARE EVALUATED MODULO M,  
C I.E., X(0,J) = X(M,J) AND X(M+1,J) = X(1,J),  
C AND X(I,0) MAY EQUAL 0, X(I,2), OR X(I,N),  
C AND X(I,N+1) MAY EQUAL 0, X(I,N-1), OR X(I,1)  
C DEPENDING ON AN INPUT PARAMETER.  
C  
C USAGE CALL GENBUN (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,  
C IERROR)  
C  
C ARGUMENTS  
C  
C ON INPUT NPEROD  
C  
C INDICATES THE VALUES THAT X(I,0) AND  
C X(I,N+1) ARE ASSUMED TO HAVE.  
C  
C = 0 IF X(I,0) = X(I,N) AND X(I,N+1) =  
C X(I,1).  
C = 1 IF X(I,0) = X(I,N+1) = 0 .  
C = 2 IF X(I,0) = 0 AND X(I,N+1) = X(I,N-1).  
C = 3 IF X(I,0) = X(I,2) AND X(I,N+1) =  
C X(I,N-1).  
C = 4 IF X(I,0) = X(I,2) AND X(I,N+1) = 0.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.  
C N MUST BE GREATER THAN 2.  
C  
C MPEROD  
C = 0 IF A(1) AND C(M) ARE NOT ZERO  
C = 1 IF A(1) = C(M) = 0  
C  
C M  
C THE NUMBER OF UNKNOWNS IN THE I-DIRECTION.  
C N MUST BE GREATER THAN 2.  
C  
C A,B,C  
C ONE-DIMENSIONAL ARRAYS OF LENGTH M THAT  
C SPECIFY THE COEFFICIENTS IN THE LINEAR  
C EQUATIONS GIVEN ABOVE. IF MPEROD = 0  
C 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  
C A(I) = C(1)  
C C(I) = C(1)  
C B(I) = B(1)  
C  
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  
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  
C  
C CONTAINS THE SOLUTION X.  
C  
C IERROR  
C AN ERROR FLAG WHICH INDICATES INVALID  
C INPUT PARAMETERS EXCEPT FOR NUMBER  
C ZERO, A SOLUTION IS NOT ATTEMPTED.  
C  
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(1) OR C(I) .NE. C(1) OR  
C B(I) .NE. B(1) FOR  
C SOME I=1,2,...,M.  
C = 7 A(1) .NE. 0 OR C(M) .NE. 0 AND  
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  
C  
C LANGUAGE FORTRAN 90  
C  
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.  
c Revised in June 2004 by John Adams using  
c 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.  
C  
C PORTABILITY FORTRAN 90 --  
C THE MACHINE DEPENDENT CONSTANT PI IS  
C DEFINED IN FUNCTION PIMACH.  
C  
C REFERENCES SWEET, R., "A CYCLIC REDUCTION ALGORITHM FOR  
C SOLVING BLOCK TRIDIAGONAL SYSTEMS OF ARBITRARY  
C DIMENSIONS," SIAM J. ON NUMER. ANAL., 14 (1977)  
C 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  
C AND, WHEN MPEROD = 1  
C  
C A(1) = C(M) = 0  
C A(M) = C(1) = 2.  
C  
C THE SOLUTION X WAS SUBSTITUTED INTO THE  
C GIVEN SYSTEM AND, USING DOUBLE PRECISION  
C A RIGHT SIDE Y WAS COMPUTED.  
C USING THIS ARRAY Y, SUBROUTINE GENBUN  
C WAS CALLED TO PRODUCE APPROXIMATE  
C SOLUTION Z. THEN RELATIVE ERROR  
C E = MAX(ABS(Z(I,J)-X(I,J)))/  
C MAX(ABS(X(I,J)))  
C WAS COMPUTED, WHERE THE TWO MAXIMA ARE TAKEN  
C OVER I=1,2,...,M AND J=1,...,N.  
C  
C THE VALUE OF E IS GIVEN IN THE TABLE  
C BELOW FOR SOME TYPICAL VALUES OF M AND N.  
C  
C M (=N) MPEROD NPEROD E  
C ------ ------ ------ ------  
C  
C 31 0 0 6.E-14  
C 31 1 1 4.E-13  
C 31 1 3 3.E-13  
C 32 0 0 9.E-14  
C 32 1 1 3.E-13  
C 32 1 3 1.E-13  
C 33 0 0 9.E-14  
C 33 1 1 4.E-13  
C 33 1 3 1.E-13  
C 63 0 0 1.E-13  
C 63 1 1 1.E-12  
C 63 1 3 2.E-13  
C 64 0 0 1.E-13  
C 64 1 1 1.E-12  
C 64 1 3 6.E-13  
C 65 0 0 2.E-13  
C 65 1 1 1.E-12  
C 65 1 3 4.E-13  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

**GNBNAUX**

C  
C file gnbnaux.txt (documentation for the FISHPACK file gnbnaux.f)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C PACKAGE GNBNAUX  
C  
C LATEST REVISION June 2004  
C  
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 SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED files fish.f, comf.f  
C  
C LANGUAGE FORTRAN 90  
C  
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.  
c Revised in June 2004 by John Adams using  
c Fortran 90 dynamically allocated work space.  
C  
C PORTABILITY FORTRAN 90  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HSTCRT**

C  
C file hstcrt.txt (documentation for the FISHPACK solver HSTCRT)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HSTCRT (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION TO THE HELMHOLTZ  
C EQUATION  
C (D/DX)(DU/DX) + (D/DY)(DU/DY) + LAMBDA\*U  
C = F(X,Y)  
C ON A STAGGERED GRID IN CARTESIAN COORDINATES.  
C  
C USAGE CALL HSTCRT (A,B,M,MBDCND,BDA,BDB,C,D  
C N,NBDCND,BDC,BDD,ELMBDA,  
C F,IDIMF,PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT  
C  
C A,B  
C THE RANGE OF X, I.E. A .LE. X .LE. B.  
C A MUST BE LESS THAN B.  
C  
C M  
C THE NUMBER OF GRID POINTS IN THE  
C INTERVAL (A,B). THE GRID POINTS  
C IN THE X-DIRECTION ARE GIVEN BY  
C X(I) = A + (I-0.5)DX FOR I=1,2,...,M  
C WHERE DX =(B-A)/M. M MUST BE GREATER  
C THAN 2.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT X = A AND X = B.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN X,  
C U(M+I,J) = U(I,J).  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C X = A AND X = B.  
C  
C = 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.  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO X IS SPECIFIED  
C AT X = A AND X = B.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO X IS SPECIFIED  
C AT X = A AND THE SOLUTION IS  
C SPECIFIED AT X = B.  
C  
C BDA  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N  
C THAT SPECIFIES THE BOUNDARY VALUES  
C (IF ANY) OF THE SOLUTION AT X = A.  
C  
C WHEN MBDCND = 1 OR 2,  
C BDA(J) = U(A,Y(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 3 OR 4,  
C BDA(J) = (D/DX)U(A,Y(J)) , J=1,2,...,N.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N  
C THAT SPECIFIES THE BOUNDARY VALUES  
C OF THE SOLUTION AT X = B.  
C  
C WHEN MBDCND = 1 OR 4  
C BDB(J) = U(B,Y(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 2 OR 3  
C BDB(J) = (D/DX)U(B,Y(J)) , J=1,2,...,N.  
C  
C C,D  
C THE RANGE OF Y, I.E. C .LE. Y .LE. D.  
C C MUST BE LESS THAN D.  
C  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE INTERVAL  
C (C,D). THE UNKNOWNS IN THE Y-DIRECTION  
C ARE GIVEN BY Y(J) = C + (J-0.5)DY,  
C J=1,2,...,N, WHERE DY = (D-C)/N.  
C N MUST BE GREATER THAN 2.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT Y = C AND Y = D.  
C  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN Y, I.E.  
C U(I,J) = U(I,N+J).  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT Y = C  
C AND Y = D.  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT Y = C  
C AND THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = D.  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = C AND Y = D.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = C AND THE SOLUTION IS SPECIFIED  
C AT Y = D.  
C  
C BDC  
C A ONE DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT Y = C.  
C  
C WHEN NBDCND = 1 OR 2,  
C BDC(I) = U(X(I),C) , I=1,2,...,M.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DY)U(X(I),C), I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDC IS A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT Y = D.  
C  
C WHEN NBDCND = 1 OR 4,  
C BDD(I) = U(X(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) = (D/DY)U(X(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDD IS A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA IS GREATER THAN 0,  
C A SOLUTION MAY NOT EXIST. HOWEVER,  
C HSTCRT WILL ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE RIGHT SIDE OF THE  
C HELMHOLTZ EQUATION. FOR I=1,2,...,M  
C AND J=1,2,...,N  
C  
C F(I,J) = F(X(I),Y(J)) .  
C  
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 HSTCRT. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F.  
C IDIMF MUST BE AT LEAST M.  
C  
C  
C ON OUTPUT 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 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. HSTCRT  
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 PROBLEM.  
C THIS COMPARISON SHOULD ALWAYS BE MADE TO  
C INSURE THAT A MEANINGFUL SOLUTION HAS BEEN  
C OBTAINED.  
C  
C IERROR  
C AN ERROR FLAG THAT INDICATES INVALID INPUT  
C PARAMETERS. EXCEPT TO NUMBERS 0 AND 6,  
C A SOLUTION IS NOT ATTEMPTED.  
C  
C = 0 NO ERROR  
C  
C = 1 A .GE. B  
C  
C = 2 MBDCND .LT. 0 OR MBDCND .GT. 4  
C  
C = 3 C .GE. D  
C  
C = 4 N .LE. 2  
C  
C = 5 NBDCND .LT. 0 OR NBDCND .GT. 4  
C  
C = 6 LAMBDA .GT. 0  
C  
C = 7 IDIMF .LT. M  
C  
C = 8 M .LE. 2  
C  
C SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HSTCRT, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
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 I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED LIBRARY fish.f,comf.f,genbun.f,gnbnaux.f,poistg.f  
C FILES  
C  
C LANGUAGE FORTRAN 90  
C  
C HISTORY WRITTEN BY ROLAND SWEET AT NCAR IN 1977.  
C RELEASED ON NCAR'S PUBLIC SOFTWARE LIBRARIES  
C 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 BOUNDARY DATA, ADJUSTS  
C THE RIGHT SIDE WHEN THE SYSTEM IS SINGULAR  
C AND CALLS EITHER POISTG OR GENBUN WHICH SOLVES  
C THE LINEAR SYSTEM OF EQUATIONS.  
C  
C TIMING FOR LARGE M AND N, THE OPERATION COUNT  
C IS ROUGHLY PROPORTIONAL TO M\*N\*LOG2(N).  
C  
C ACCURACY THE SOLUTION PROCESS EMPLOYED RESULTS IN A  
C LOSS OF NO MORE THAN FOUR SIGNIFICANT DIGITS  
C FOR N AND M AS LARGE AS 64. MORE DETAILED  
C INFORMATION ABOUT ACCURACY CAN BE FOUND IN  
C THE DOCUMENTATION FOR PACKAGE POISTG WHICH  
C SOLVES THE FINITE DIFFERENCE EQUATIONS.  
C  
C REFERENCES U. SCHUMANN AND R. SWEET,"A DIRECT METHOD  
C FOR THE SOLUTION OF POISSON'S EQUATION WITH  
C BOUNDARY CONDITIONS ON A STAGGERED GRID OF  
C ARBITRARY SIZE," J. COMP. PHYS. 20(1976),  
C PP. 171-182.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HSTCSP**

C  
C file hstcsp.txt (documentation for the FISHPACK solver HSTCSP)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HSTCSP (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,  
C + BDD,ELMBDA,F,IDIMF,PERTRB,IERROR,W)  
C  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION ON A STAGGERED  
C GRID TO THE MODIFIED HELMHOLTZ EQUATION IN  
C SPHERICAL COORDINATES ASSUMING AXISYMMETRY  
C (NO DEPENDENCE ON LONGITUDE).  
C  
C THE EQUATION IS  
C  
C (1/R\*\*2)(D/DR)(R\*\*2(DU/DR)) +  
C 1/(R\*\*2\*SIN(THETA))(D/DTHETA)  
C (SIN(THETA)(DU/DTHETA)) +  
C (LAMBDA/(R\*SIN(THETA))\*\*2)U = F(THETA,R)  
C  
C WHERE THETA IS COLATITUDE AND R IS THE  
C RADIAL COORDINATE. THIS TWO-DIMENSIONAL  
C MODIFIED HELMHOLTZ EQUATION RESULTS FROM  
C THE FOURIER TRANSFORM OF THE THREE-  
C DIMENSIONAL POISSON EQUATION.  
C  
C  
C USAGE CALL HSTCSP (INTL,A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR,W)  
C  
C ARGUMENTS  
C ON INPUT INTL  
C  
C = 0 ON INITIAL ENTRY TO HSTCSP OR IF ANY  
C OF THE ARGUMENTS C, D, N, OR NBDCND  
C ARE CHANGED FROM A PREVIOUS CALL  
C  
C = 1 IF C, D, N, AND NBDCND ARE ALL  
C UNCHANGED FROM PREVIOUS CALL TO HSTCSP  
C  
C 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  
C HAS BEEN MADE THEN SUBSEQUENT SOLUTIONS  
C CORRESPONDING TO DIFFERENT F, BDA, BDB,  
C BDC, AND BDD CAN BE OBTAINED FASTER WITH  
C INTL = 1 SINCE INITIALIZATION IS NOT  
C REPEATED.  
C  
C A,B  
C THE RANGE OF THETA (COLATITUDE),  
C I.E. A .LE. THETA .LE. B. A  
C MUST BE LESS THAN B AND A MUST BE  
C NON-NEGATIVE. A AND B ARE IN RADIANS.  
C A = 0 CORRESPONDS TO THE NORTH POLE AND  
C B = PI CORRESPONDS TO THE SOUTH POLE.  
C  
C \* \* \* IMPORTANT \* \* \*  
C  
C 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 IS EQUAL TO PI IN THIS PROGRAM, PERMITTING  
C SEVERAL TESTS OF THE INPUT PARAMETERS THAT  
C OTHERWISE WOULD NOT BE POSSIBLE.  
C  
C \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C M  
C THE NUMBER OF GRID POINTS IN THE INTERVAL  
C (A,B). THE GRID POINTS IN THE THETA-  
C DIRECTION ARE GIVEN BY  
C THETA(I) = A + (I-0.5)DTHETA  
C FOR I=1,2,...,M WHERE DTHETA =(B-A)/M.  
C M MUST BE GREATER THAN 4.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT THETA = A AND THETA = B.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THETA = B.  
C (SEE NOTES 1, 2 BELOW)  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = B  
C (SEE NOTES 1, 2 BELOW).  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = A (SEE NOTES 1, 2 BELOW)  
C AND THETA = B.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED AT  
C THETA = A (SEE NOTES 1, 2 BELOW) AND  
C THE SOLUTION IS SPECIFIED AT THETA = B.  
C  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THE SOLUTION IS  
C SPECIFIED AT THETA = B.  
C (SEE NOTE 2 BELOW)  
C  
C = 6 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = B  
C (SEE NOTE 2 BELOW).  
C  
C = 7 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THE SOLUTION IS  
C UNSPECIFIED AT THETA = B = PI.  
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  
C THETA = B = PI.  
C  
C = 9 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THETA = B = PI.  
C  
C NOTE 1:  
C IF A = 0, DO NOT USE MBDCND = 1,2,3,4,7  
C OR 8, BUT INSTEAD USE MBDCND = 5, 6, OR 9.  
C  
C NOTE 2:  
C IF B = PI, DO NOT USE MBDCND = 1,2,3,4,5,  
C OR 6, BUT INSTEAD USE MBDCND = 7, 8, OR 9.  
C  
C NOTE 3:  
C WHEN A = 0 AND/OR B = PI THE ONLY  
C MEANINGFUL BOUNDARY CONDITION IS  
C DU/DTHETA = 0. SEE D. GREENSPAN,  
C 'NUMERICAL ANALYSIS OF ELLIPTIC  
C BOUNDARY VALUE PROBLEMS,'  
C HARPER AND ROW, 1965, CHAPTER 5.)  
C  
C BDA  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES (IF ANY) OF  
C THE SOLUTION AT THETA = A.  
C  
C WHEN MBDCND = 1, 2, OR 7,  
C BDA(J) = U(A,R(J)), J=1,2,...,N.  
C  
C WHEN MBDCND = 3, 4, OR 8,  
C BDA(J) = (D/DTHETA)U(A,R(J)), J=1,2,...,N.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDA IS A  
C DUMMY VARIABLE.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT THETA = B.  
C  
C WHEN MBDCND = 1, 4, OR 5,  
C BDB(J) = U(B,R(J)), J=1,2,...,N.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DTHETA)U(B,R(J)), J=1,2,...,N.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDB IS  
C A DUMMY VARIABLE.  
C  
C C,D  
C THE RANGE OF R , I.E. C .LE. R .LE. D.  
C C MUST BE LESS THAN D AND NON-NEGATIVE.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE INTERVAL  
C (C,D). THE UNKNOWNS IN THE R-DIRECTION  
C ARE GIVEN BY R(J) = C + (J-0.5)DR,  
C J=1,2,...,N, WHERE DR = (D-C)/N.  
C N MUST BE GREATER THAN 4.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT R = C AND R = D.  
C  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C R = C AND R = D.  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C R = C AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO R IS  
C SPECIFIED AT R = D. (SEE NOTE 1 BELOW)  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = C AND R = D.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C 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  
C R = C = 0 (SEE NOTE 2 BELOW) AND THE  
C SOLUTION IS SPECIFIED AT R = D.  
C  
C = 6 IF THE SOLUTION IS UNSPECIFIED AT  
C R = C = 0 (SEE NOTE 2 BELOW)  
C AND THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = D.  
C  
C NOTE 1:  
C IF C = 0 AND MBDCND = 3,6,8 OR 9, THE  
C SYSTEM OF EQUATIONS TO BE SOLVED IS  
C SINGULAR. THE UNIQUE SOLUTION IS  
C DETERMINED BY EXTRAPOLATION TO THE  
C SPECIFICATION OF U(THETA(1),C).  
C BUT IN THESE CASES THE RIGHT SIDE OF THE  
C SYSTEM WILL BE PERTURBED BY THE CONSTANT  
C PERTRB.  
C  
C NOTE 2:  
C NBDCND = 5 OR 6 CANNOT BE USED WITH  
C MBDCND =1, 2, 4, 5, OR 7  
C (THE FORMER INDICATES THAT THE SOLUTION IS  
C UNSPECIFIED AT R = 0; THE LATTER INDICATES  
C SOLUTION IS SPECIFIED).  
C USE INSTEAD NBDCND = 1 OR 2.  
C  
C BDC  
C A ONE DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT R = C. WHEN NBDCND = 1 OR 2,  
C BDC(I) = U(THETA(I),C), I=1,2,...,M.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DR)U(THETA(I),C), I=1,2,...,M.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC IS  
C A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT R = D. WHEN NBDCND = 1 OR 4,  
C BDD(I) = U(THETA(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) = (D/DR)U(THETA(I),D), I=1,2,...,M.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDD IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE MODIFIED  
C HELMHOLTZ EQUATION. IF LAMBDA IS GREATER  
C THAN 0, A SOLUTION MAY NOT EXIST.  
C HOWEVER, HSTCSP WILL ATTEMPT TO FIND A  
C SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE RIGHT SIDE OF THE MODIFIED  
C HELMHOLTZ EQUATION. FOR I=1,2,...,M AND  
C J=1,2,...,N  
C  
C F(I,J) = F(THETA(I),R(J)) .  
C  
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 HSTCSP. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F.  
C IDIMF MUST BE AT LEAST M.  
C  
C W  
c A fortran 90 derived TYPE (fishworkspace) variable  
c that must be declared by the user. The first  
c two declarative statements in the user program  
c calling HSTCSP must be:  
c  
c USE fish  
c  
c The declarative statement  
c  
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 HSTCSP. The second statement  
c declares a derived type variable (defined in  
c the module "fish.f") which is used internally  
c in BLKTRI 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  
c 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 (IFLG=0) call to HSTCSP. These  
c must be preserved on non-initial calls (INTL=1)  
c to HSTCSP. This eliminates redundant calculations  
c and saves compute time.  
c \*\*\*\* IMPORTANT! The user program calling HSTCSP should  
c include the statement:  
c  
c CALL FISHFIN(W)  
C  
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  
c could result in serious memory leakage.  
C  
C   
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),R(J)) FOR I=1,2,..,M, J=1,2,...,N.  
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  
C SOLUTION EXISTS. HSTCSP THEN COMPUTES THIS  
C 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.  
C THIS COMPARISON SHOULD ALWAYS BE MADE TO  
C INSURE 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 10,  
C A SOLUTION IS NOT ATTEMPTED.  
C  
C = 0 NO ERROR  
C  
C = 1 A .LT. 0 OR B .GT. PI  
C  
C = 2 A .GE. B  
C  
C = 3 MBDCND .LT. 1 OR MBDCND .GT. 9  
C  
C = 4 C .LT. 0  
C  
C = 5 C .GE. D  
C  
C = 6 NBDCND .LT. 1 OR NBDCND .GT. 6  
C  
C = 7 N .LT. 5  
C  
C = 8 NBDCND = 5 OR 6 AND  
C MBDCND = 1, 2, 4, 5, OR 7  
C  
C = 9 C .GT. 0 AND NBDCND .GE. 5  
C  
C = 10 ELMBDA .GT. 0  
C  
C = 11 IDIMF .LT. M  
C  
C = 12 M .LT. 5  
C  
C = 13 A = 0 AND MBDCND =1,2,3,4,7 OR 8  
C  
C = 14 B = PI AND MBDCND .LE. 6  
C  
C = 15 A .GT. 0 AND MBDCND = 5, 6, OR 9  
C  
C = 16 B .LT. PI AND MBDCND .GE. 7  
C  
C = 17 LAMBDA .NE. 0 AND NBDCND .GE. 5  
C  
C SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HSTCSP,  
C THE USER SHOULD TEST IERROR AFTER THE CALL.  
C  
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   
C W  
c The derived type (fishworkspace) variable W  
c contains real and complex values that must not  
C be destroyed if HSTCSP is called again with  
C IFLG=1.  
C   
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED LIBRARY fish.f,blktri.f,comf.f  
C FILES  
C  
C LANGUAGE FORTRAN 90  
C  
C HISTORY WRITTEN BY ROLAND SWEET AT NCAR IN 1977.  
C RELEASED ON NCAR'S PUBLIC SOFTWARE LIBRARIES  
C IN JANUARY 1980. Revised by John Adams in June  
C 2004 using Fortan 90 dynamically allocated work  
c space and derived data types to eliminate mixed  
c mode conflicts in the earlier versions.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THIS SUBROUTINE DEFINES THE FINITE-DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, ADJUSTS  
C THE RIGHT SIDE WHEN THE SYSTEM IS SINGULAR  
C AND CALLS BLKTRI WHICH SOLVES THE LINEAR  
C SYSTEM OF EQUATIONS.  
C  
C  
C TIMING FOR LARGE M AND N, THE OPERATION COUNT IS  
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.  
C MORE DETAILED INFORMATION ABOUT ACCURACY  
C CAN BE FOUND IN THE DOCUMENTATION FOR  
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  
C EQUATIONS",  
C SIAM J. NUMER. ANAL. 11(1974), PP. 1136-1150.  
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 ARBITRARY SIZE," J. COMP. PHYS. 20(1976),  
C PP. 171-182.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HSTCYL**

C  
C file hstcyl.txt (documentation for the FISHPACK solver HSTCYL)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HSTCYL (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION ON A STAGGERED  
C GRID TO THE MODIFIED HELMHOLTZ EQUATION  
C IN CYLINDRICAL COORDINATES. THIS EQUATION  
C  
C (1/R)(D/DR)(R(DU/DR)) + (D/DZ)(DU/DZ)  
C  
C + LAMBDA\*(1/R\*\*2)\*U = F(R,Z)  
C  
C IS A TWO-DIMENSIONAL MODIFIED HELMHOLTZ  
C EQUATION RESULTING FROM THE FOURIER TRANSFORM  
C OF A THREE-DIMENSIONAL POISSON EQUATION.  
C  
C USAGE CALL HSTCYL (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT A,B  
C  
C THE RANGE OF R, I.E. A .LE. R .LE. B.  
C A MUST BE LESS THAN B AND A MUST BE  
C BE NON-NEGATIVE.  
C  
C M  
C THE NUMBER OF GRID POINTS IN THE INTERVAL  
C (A,B). THE GRID POINTS IN THE R-DIRECTION  
C R-DIRECTION ARE GIVEN BY  
C R(I) = A + (I-0.5)DR FOR I=1,2,...,M  
C 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  
C = 1 IF THE SOLUTION IS SPECIFIED AT R = A  
C (SEE NOTE BELOW) AND R = B.  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT R = A  
C (SEE NOTE BELOW) AND THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO R IS  
C SPECIFIED AT R = B.  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND R = B.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND THE  
C SOLUTION IS SPECIFIED AT R = B.  
C  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C 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 SOLUTION WITH RESPECT TO R IS SPECIFIED  
C AT R = B.  
C  
C NOTE:  
C IF A = 0, DO NOT USE MBDCND = 1,2,3, OR 4,  
C BUT INSTEAD USE MBDCND = 5 OR 6.  
C THE RESULTING APPROXIMATION GIVES THE ONLY  
C MEANINGFUL BOUNDARY CONDITION,  
C I.E. DU/DR = 0.  
C (SEE D. GREENSPAN, 'INTRODUCTORY NUMERICAL  
C ANALYSIS OF ELLIPTIC BOUNDARY VALUE  
C PROBLEMS,' HARPER AND ROW, 1965, CHAPTER 5.)  
C  
C BDA  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES (IF ANY)  
C OF THE SOLUTION AT R = A.  
C  
C WHEN MBDCND = 1 OR 2,  
C BDA(J) = U(A,Z(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 3 OR 4,  
C BDA(J) = (D/DR)U(A,Z(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 5 OR 6, BDA IS A DUMMY  
C VARIABLE.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT R = B.  
C  
C WHEN MBDCND = 1,4,OR 5,  
C BDB(J) = U(B,Z(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DR)U(B,Z(J)) , J=1,2,...,N.  
C  
C C,D  
C THE RANGE OF Z, I.E. C .LE. Z .LE. D.  
C C MUST BE LESS THAN D.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE INTERVAL  
C (C,D). THE UNKNOWNS IN THE Z-DIRECTION  
C ARE GIVEN BY Z(J) = C + (J-0.5)DZ,  
C J=1,2,...,N, WHERE DZ = (D-C)/N.  
C N MUST BE GREATER THAN 2.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT Z = C AND Z = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN Z, I.E.  
C U(I,J) = U(I,N+J).  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT Z = C  
C AND Z = D.  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT Z = C  
C AND THE DERIVATIVE OF THE SOLUTION WITH  
C RESPECT TO Z IS SPECIFIED AT Z = D.  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION WITH  
C RESPECT TO Z IS SPECIFIED AT Z = C  
C AND Z = D.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION WITH  
C RESPECT TO Z IS SPECIFIED AT Z = C AND  
C THE SOLUTION IS SPECIFIED AT Z = D.  
C  
C BDC  
C A ONE DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT Z = C.  
C  
C WHEN NBDCND = 1 OR 2,  
C BDC(I) = U(R(I),C) , I=1,2,...,M.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DZ)U(R(I),C), I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDC IS A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT Z = D.  
C  
C WHEN NBDCND = 1 OR 4,  
C BDD(I) = U(R(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) = (D/DZ)U(R(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDD IS A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE MODIFIED  
C HELMHOLTZ EQUATION. IF LAMBDA IS GREATER  
C THAN 0, A SOLUTION MAY NOT EXIST.  
C HOWEVER, HSTCYL WILL ATTEMPT TO FIND A  
C SOLUTION. LAMBDA MUST BE ZERO WHEN  
C MBDCND = 5 OR 6.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE RIGHT SIDE OF THE  
C MODIFIED HELMHOLTZ EQUATION.  
C FOR I=1,2,...,M AND J=1,2,...,N  
C F(I,J) = F(R(I),Z(J)) .  
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 HSTCYL. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F. IDIMF MUST  
C BE AT LEAST M.  
C  
C ON OUTPUT  
C  
C F  
C CONTAINS THE SOLUTION U(I,J) OF THE FINITE  
C DIFFERENCE APPROXIMATION FOR THE GRID POINT  
C (R(I),Z(J)) FOR I=1,2,...,M, J=1,2,...,N.  
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  
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  
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.  
C THIS COMPARISON SHOULD ALWAYS BE MADE TO  
C INSURE THAT A MEANINGFUL SOLUTION HAS BEEN  
C OBTAINED.  
C  
C IERROR  
C 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  
C  
C = 1 A .LT. 0  
C  
C = 2 A .GE. B  
C  
C = 3 MBDCND .LT. 1 OR MBDCND .GT. 6  
C  
C = 4 C .GE. D  
C  
C = 5 N .LE. 2  
C  
C = 6 NBDCND .LT. 0 OR NBDCND .GT. 4  
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  
C  
C = 10 IDIMF .LT. M  
C  
C = 11 LAMBDA .GT. 0  
C  
C = 12 A=0, MBDCND .GE. 5, ELMBDA .NE. 0  
C  
C SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HSTCYL, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
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 I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED LIBRARY fish.f,comf.f,genbun.f,gnbnaux.f,poistg.f  
C FILES  
C  
C LANGUAGE FORTRAN 90  
C  
C HISTORY WRITTEN BY ROLAND SWEET AT NCAR IN 1977.  
C RELEASED ON NCAR'S PUBLIC SOFTWARE LIBRARIES  
C 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 BOUNDARY DATA, ADJUSTS  
C THE RIGHT SIDE WHEN THE SYSTEM IS SINGULAR AND  
C CALLS EITHER POISTG OR GENBUN WHICH SOLVES THE  
C LINEAR SYSTEM OF EQUATIONS.  
C  
C TIMING FOR LARGE M AND N, THE OPERATION COUNT  
C IS ROUGHLY PROPORTIONAL TO M\*N\*LOG2(N).  
C  
C ACCURACY THE SOLUTION PROCESS RESULTS IN A LOSS  
C OF NO MORE THAN FOUR SIGNIFICANT DIGITS  
C FOR N AND M AS LARGE AS 64.  
C MORE DETAILED INFORMATION ABOUT ACCURACY  
C CAN BE FOUND IN THE DOCUMENTATION FOR  
C SUBROUTINE POISTG WHICH IS THE ROUTINE THAT  
C ACTUALLY SOLVES THE FINITE DIFFERENCE  
C EQUATIONS.  
C  
C REFERENCES 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 ARBITRARY SIZE," J. COMP. PHYS. 20(1976),  
C PP. 171-182.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HSTPLR**

C  
C file hstplr.txt (documentation for the FISHPACK solver HSTPLR)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HSTPLR (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION ON A STAGGERED  
C GRID TO THE HELMHOLTZ EQUATION IN POLAR  
C COORDINATES. THE EQUATION IS  
C  
C (1/R)(D/DR)(R(DU/DR)) +  
C (1/R\*\*2)(D/DTHETA)(DU/DTHETA) +  
C LAMBDA\*U = F(R,THETA)  
C  
C USAGE CALL HSTPLR (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,  
C IDIMF,PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT A,B  
C  
C THE RANGE OF R, I.E. A .LE. R .LE. B.  
C A MUST BE LESS THAN B AND A MUST BE  
C NON-NEGATIVE.  
C  
C M  
C THE NUMBER OF GRID POINTS IN THE INTERVAL  
C (A,B). THE GRID POINTS IN THE R-DIRECTION  
C 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.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT R = A AND R = B.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT R = A  
C AND R = B.  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT R = A  
C AND THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT R = B.  
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  
C WITH RESPECT TO R IS SPECIFIED AT  
C SPECIFIED AT R = A (SEE NOTE 2 BELOW)  
C AND THE SOLUTION IS SPECIFIED AT R = B.  
C  
C  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C 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 SOLUTION WITH RESPECT TO R IS SPECIFIED  
C AT R = B.  
C  
C NOTE 1:  
C IF A = 0, MBDCND = 2, AND NBDCND = 0 OR 3,  
C THE SYSTEM OF EQUATIONS TO BE SOLVED IS  
C SINGULAR. THE UNIQUE SOLUTION IS  
C IS DETERMINED BY EXTRAPOLATION TO THE  
C SPECIFICATION OF U(0,THETA(1)).  
C BUT IN THIS CASE THE RIGHT SIDE OF THE  
C SYSTEM WILL BE PERTURBED BY THE CONSTANT  
C PERTRB.  
C  
C NOTE 2:  
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 THAT  
C SPECIFIES THE BOUNDARY VALUES (IF ANY) OF  
C THE SOLUTION AT R = A.  
C  
C WHEN MBDCND = 1 OR 2,  
C BDA(J) = U(A,THETA(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 3 OR 4,  
C BDA(J) = (D/DR)U(A,THETA(J)) ,  
C J=1,2,...,N.  
C  
C WHEN MBDCND = 5 OR 6, BDA IS A DUMMY  
C VARIABLE.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT R = B.  
C  
C WHEN MBDCND = 1,4, OR 5,  
C BDB(J) = U(B,THETA(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DR)U(B,THETA(J)) ,  
C J=1,2,...,N.  
C  
C C,D  
C THE RANGE OF THETA, I.E. C .LE. THETA .LE. D.  
C C MUST BE LESS THAN D.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE INTERVAL  
C (C,D). THE UNKNOWNS IN THE THETA-  
C DIRECTION ARE GIVEN BY THETA(J) = C +  
C (J-0.5)DT, J=1,2,...,N, WHERE  
C DT = (D-C)/N. N MUST BE GREATER THAN 2.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT THETA = C AND THETA = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN THETA,  
C I.E. U(I,J) = U(I,N+J).  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = C AND THETA = D  
C (SEE NOTE BELOW).  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C THETA = C AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = D  
C (SEE NOTE BELOW).  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = C AND THETA = D.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = C AND THE SOLUTION IS  
C SPECIFIED AT THETA = D  
C (SEE NOTE BELOW).  
C  
C NOTE:  
C WHEN NBDCND = 1, 2, OR 4, DO NOT USE  
C MBDCND = 5 OR 6 (THE FORMER INDICATES THAT  
C THE SOLUTION IS SPECIFIED AT R = 0; THE  
C LATTER INDICATES THE SOLUTION IS UNSPECIFIED  
C AT R = 0). USE INSTEAD MBDCND = 1 OR 2.  
C  
C BDC  
C A ONE DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT THETA = C.  
C  
C WHEN NBDCND = 1 OR 2,  
C BDC(I) = U(R(I),C) , I=1,2,...,M.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DTHETA)U(R(I),C),  
C I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDC IS A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT THETA = D.  
C  
C WHEN NBDCND = 1 OR 4,  
C BDD(I) = U(R(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) =(D/DTHETA)U(R(I),D), I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDD IS A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA IS GREATER THAN 0,  
C A SOLUTION MAY NOT EXIST. HOWEVER, HSTPLR  
C WILL ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE RIGHT SIDE OF THE HELMHOLTZ  
C EQUATION.  
C  
C FOR I=1,2,...,M AND J=1,2,...,N  
C F(I,J) = F(R(I),THETA(J)) .  
C  
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 HSTPLR. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F.  
C IDIMF MUST BE AT LEAST M.  
C  
C  
C ON OUTPUT  
C  
C F  
C 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,...,M,  
C J=1,2,...,N.  
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  
C SOLUTION EXISTS. HSTPLR THEN COMPUTES THIS  
C 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.  
C THIS COMPARISON SHOULD ALWAYS BE MADE TO  
C INSURE THAT A MEANINGFUL SOLUTION HAS BEEN  
C OBTAINED.  
C  
C IERROR  
C 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  
C  
C = 1 A .LT. 0  
C  
C = 2 A .GE. B  
C  
C = 3 MBDCND .LT. 1 OR MBDCND .GT. 6  
C  
C = 4 C .GE. D  
C  
C = 5 N .LE. 2  
C  
C = 6 NBDCND .LT. 0 OR NBDCND .GT. 4  
C  
C = 7 A = 0 AND MBDCND = 3 OR 4  
C  
C = 8 A .GT. 0 AND MBDCND .GE. 5  
C  
C = 9 MBDCND .GE. 5 AND NBDCND .NE. 0 OR 3  
C  
C = 10 IDIMF .LT. M  
C  
C = 11 LAMBDA .GT. 0  
C  
C = 12 M .LE. 2  
C  
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 SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HSTPLR, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
C  
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  
C HISTORY WRITTEN BY ROLAND SWEET AT NCAR IN 1977.  
C RELEASED ON NCAR'S PUBLIC SOFTWARE LIBRARIES  
C 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-  
C DIFFERENCE EQUATIONS, INCORPORATES BOUNDARY  
C DATA, ADJUSTS THE RIGHT SIDE WHEN THE SYSTEM  
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  
C 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.  
C 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  
C REFERENCES U. SCHUMANN AND R. SWEET, "A DIRECT METHOD  
C FOR THE SOLUTION OF POISSON'S EQUATION WITH  
C NEUMANN BOUNDARY CONDITIONS ON A STAGGERED  
C GRID OF ARBITRARY SIZE," J. COMP. PHYS.  
C 20(1976), PP. 171-182.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HSTSSP**

C  
C file hstssp.txt (documentation for the FISHPACK solver HSTSSP)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HSTSSP (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION ON A STAGGERED GRID  
C TO THE HELMHOLTZ EQUATION IN SPHERICAL  
C COORDINATES AND ON THE SURFACE OF THE UNIT  
C SPHERE (RADIUS OF 1). THE EQUATION IS  
C  
C (1/SIN(THETA))(D/DTHETA)(SIN(THETA)  
C (DU/DTHETA)) + (1/SIN(THETA)\*\*2)  
C (D/DPHI)(DU/DPHI) + LAMBDA\*U = F(THETA,PHI)  
C  
C WHERE THETA IS COLATITUDE AND PHI IS  
C LONGITUDE.  
C  
C USAGE CALL HSTSSP (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR)  
C  
C  
C ARGUMENTS  
C ON INPUT  
C  
C A,B  
C THE RANGE OF THETA (COLATITUDE),  
C I.E. A .LE. THETA .LE. B.  
C A MUST BE LESS THAN B AND A MUST BE  
C NON-NEGATIVE. A AND B ARE IN RADIANS.  
C A = 0 CORRESPONDS TO THE NORTH POLE AND  
C B = PI CORRESPONDS TO THE SOUTH POLE.  
C  
C  
C \* \* \* IMPORTANT \* \* \*  
C  
C IF B IS EQUAL TO PI, THEN B MUST BE  
C COMPUTED USING THE STATEMENT  
C B = PIMACH(DUM)  
C  
C THIS INSURES THAT B IN THE USER"S PROGRAM  
C IS EQUAL TO PI IN THIS PROGRAM WHICH  
C PERMITS SEVERAL TESTS OF THE INPUT  
C PARAMETERS THAT OTHERWISE WOULD NOT BE  
C POSSIBLE.  
C  
C \* \* \* \* \* \* \* \* \* \* \* \*  
C M  
C THE NUMBER OF GRID POINTS IN THE INTERVAL  
C (A,B). THE GRID POINTS IN THE THETA  
C DIRECTION ARE GIVEN BY  
C THETA(I) = A + (I-0.5)DTHETA  
C FOR I=1,2,...,M WHERE DTHETA =(B-A)/M.  
C M MUST BE GREATER THAN 2.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT THETA = A AND THETA = B.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THETA = B.  
C (SEE NOTE 3 BELOW)  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = B  
C (SEE NOTES 2 AND 3 BELOW).  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = A  
C (SEE NOTES 1, 2 BELOW) AND THETA = B.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = A  
C (SEE NOTES 1 AND 2 BELOW) AND THE  
C SOLUTION IS SPECIFIED AT THETA = B.  
C  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THE SOLUTION IS  
C SPECIFIED AT THETA = B.  
C (SEE NOTE 3 BELOW)  
C  
C = 6 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO THETA  
C IS SPECIFIED AT THETA = B  
C (SEE NOTE 2 BELOW).  
C  
C = 7 IF THE SOLUTION IS SPECIFIED AT  
C THETA = A AND THE SOLUTION IS  
C UNSPECIFIED AT THETA = B = PI.  
C (SEE NOTE 3 BELOW)  
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  
C THETA = B = PI.  
C  
C = 9 IF THE SOLUTION IS UNSPECIFIED AT  
C THETA = A = 0 AND THETA = B = PI.  
C  
C NOTE 1:  
C IF A = 0, DO NOT USE MBDCND = 3, 4, OR 8,  
C BUT INSTEAD USE MBDCND = 5, 6, OR 9.  
C  
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  
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 BUT IN THESE CASES THE RIGHT SIDE OF THE  
C SYSTEM WILL BE PERTURBED BY THE CONSTANT  
C PERTRB.  
C  
C BDA  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES (IF ANY) OF  
C THE SOLUTION AT THETA = A.  
C  
C WHEN MBDCND = 1, 2, OR 7,  
C BDA(J) = U(A,PHI(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 3, 4, OR 8,  
C BDA(J) = (D/DTHETA)U(A,PHI(J)) ,  
C J=1,2,...,N.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE,  
C BDA IS A DUMMY VARIABLE.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT THETA = B.  
C  
C WHEN MBDCND = 1,4, OR 5,  
C BDB(J) = U(B,PHI(J)) , J=1,2,...,N.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DTHETA)U(B,PHI(J)) ,  
C J=1,2,...,N.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDB IS  
C A DUMMY VARIABLE.  
C  
C C,D  
C THE RANGE OF PHI (LONGITUDE),  
C I.E. C .LE. PHI .LE. D.  
C C MUST BE LESS THAN D. IF D-C = 2\*PI,  
C PERIODIC BOUNDARY CONDITIONS ARE USUALLY  
C USUALLY PRESCRIBED.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE INTERVAL  
C (C,D). THE UNKNOWNS IN THE PHI-DIRECTION  
C ARE GIVEN BY PHI(J) = C + (J-0.5)DPHI,  
C J=1,2,...,N, WHERE DPHI = (D-C)/N.  
C N MUST BE GREATER THAN 2.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT PHI = C AND PHI = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN PHI,  
C I.E. U(I,J) = U(I,N+J).  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C PHI = C AND PHI = D  
C (SEE NOTE BELOW).  
C  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C PHI = C AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO PHI IS  
C SPECIFIED AT PHI = D  
C (SEE NOTE BELOW).  
C  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO PHI IS SPECIFIED  
C AT PHI = C AND PHI = D.  
C  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO PHI IS SPECIFIED  
C AT PHI = C AND THE SOLUTION IS  
C SPECIFIED AT PHI = D  
C (SEE NOTE BELOW).  
C  
C NOTE:  
C WHEN NBDCND = 1, 2, OR 4, DO NOT USE  
C MBDCND = 5, 6, 7, 8, OR 9  
C (THE FORMER INDICATES THAT THE SOLUTION  
C IS SPECIFIED AT A POLE; THE LATTER  
C INDICATES THE SOLUTION IS UNSPECIFIED).  
C USE INSTEAD MBDCND = 1 OR 2.  
C  
C BDC  
C A ONE DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT PHI = C.  
C  
C WHEN NBDCND = 1 OR 2,  
C BDC(I) = U(THETA(I),C) , I=1,2,...,M.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DPHI)U(THETA(I),C),  
C I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDC IS A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M THAT  
C SPECIFIES THE BOUNDARY VALUES OF THE  
C SOLUTION AT PHI = D.  
C  
C WHEN NBDCND = 1 OR 4,  
C BDD(I) = U(THETA(I),D) , I=1,2,...,M.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) = (D/DPHI)U(THETA(I),D) ,  
C I=1,2,...,M.  
C  
C WHEN NBDCND = 0, BDD IS A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA IS GREATER THAN 0,  
C A SOLUTION MAY NOT EXIST. HOWEVER,  
C HSTSSP WILL ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE RIGHT SIDE OF THE  
C HELMHOLTZ EQUATION.  
C FOR I=1,2,...,M AND J=1,2,...,N  
C  
C F(I,J) = F(THETA(I),PHI(J)) .  
C  
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.  
C  
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  
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  
C SOLUTION EXISTS. HSTSSP 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.  
C 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.  
C  
C IERROR  
C AN ERROR FLAG THAT INDICATES INVALID INPUT  
C PARAMETERS. EXCEPT TO NUMBERS 0 AND 14,  
C A SOLUTION IS NOT ATTEMPTED.  
C  
C = 0 NO ERROR  
C  
C = 1 A .LT. 0 OR B .GT. PI  
C  
C = 2 A .GE. B  
C  
C = 3 MBDCND .LT. 1 OR MBDCND .GT. 9  
C  
C = 4 C .GE. D  
C  
C = 5 N .LE. 2  
C  
C = 6 NBDCND .LT. 0 OR NBDCND .GT. 4  
C  
C = 7 A .GT. 0 AND MBDCND = 5, 6, OR 9  
C  
C = 8 A = 0 AND MBDCND = 3, 4, OR 8  
C  
C = 9 B .LT. PI AND MBDCND .GE. 7  
C  
C = 10 B = PI AND MBDCND = 2,3, OR 6  
C  
C = 11 MBDCND .GE. 5 AND NDBCND = 1, 2, OR 4  
C  
C = 12 IDIMF .LT. M  
C  
C = 13 M .LE. 2  
C  
C = 14 LAMBDA .GT. 0  
C  
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 SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HSTSSP, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
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  
C HISTORY WRITTEN BY ROLAND SWEET AT NCAR IN 1977.  
C RELEASED ON NCAR'S PUBLIC SOFTWARE LIBRARIES  
C 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-  
C DIFFERENCE EQUATIONS, INCORPORATES BOUNDARY  
C DATA, ADJUSTS THE RIGHT SIDE WHEN THE SYSTEM  
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  
C 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.  
C 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  
C REFERENCES U. SCHUMANN AND R. SWEET,"A DIRECT METHOD  
C FOR THE SOLUTION OF POISSON'S EQUATION WITH  
C NEUMANN BOUNDARY CONDITIONS ON A STAGGERED  
C GRID OF ARBITRARY SIZE," J. COMP. PHYS.  
C 20(1976), PP. 171-182.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HW3CRT**

C  
C file hw3crt.txt (documentation for the FISHPACK solver HW3CRT)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HW3CRT (XS,XF,L,LBDCND,BDXS,BDXF,YS,YF,M,MBDCND,BDYS,  
C + BDYF,ZS,ZF,N,NBDCND,BDZS,BDZF,ELMBDA,LDIMF,  
C + MDIMF,F,PERTRB,IERROR)  
C  
C  
C DIMENSION OF BDXS(MDIMF,N+1), BDXF(MDIMF,N+1),  
C ARGUMENTS BDYS(LDIMF,N+1), BDYF(LDIMF,N+1),  
C BDZS(LDIMF,M+1), BDZF(LDIMF,M+1),  
C F(LDIMF,MDIMF,N+1)  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION TO THE HELMHOLTZ  
C EQUATION IN CARTESIAN COORDINATES. THIS  
C EQUATION IS  
C  
C (D/DX)(DU/DX) + (D/DY)(DU/DY) +  
C (D/DZ)(DU/DZ) + LAMBDA\*U = F(X,Y,Z) .  
C  
C USAGE CALL HW3CRT (XS,XF,L,LBDCND,BDXS,BDXF,YS,YF,M,  
C MBDCND,BDYS,BDYF,ZS,ZF,N,NBDCND,  
C BDZS,BDZF,ELMBDA,LDIMF,MDIMF,F,  
C PERTRB,IERROR)  
C  
C ARGUMENTS  
C  
C ON INPUT XS,XF  
C  
C THE RANGE OF X, I.E. XS .LE. X .LE. XF .  
C XS MUST BE LESS THAN XF.  
C  
C L  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (XS,XF) IS SUBDIVIDED.  
C HENCE, THERE WILL BE L+1 GRID POINTS  
C IN THE X-DIRECTION GIVEN BY  
C 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  
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.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C X = XS AND THE 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  
C X = XS AND THE SOLUTION IS SPECIFIED  
C AT X=XF.  
C  
C BDXS  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO X AT X = XS.  
C  
C WHEN LBDCND = 3 OR 4,  
C  
C BDXS(J,K) = (D/DX)U(XS,Y(J),Z(K)),  
C J=1,2,...,M+1, K=1,2,...,N+1.  
C  
C WHEN LBDCND HAS ANY OTHER VALUE, BDXS  
C IS A DUMMY VARIABLE. BDXS MUST BE  
C DIMENSIONED AT LEAST (M+1)\*(N+1).  
C  
C BDXF  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO X AT X = XF.  
C  
C WHEN LBDCND = 2 OR 3,  
C  
C BDXF(J,K) = (D/DX)U(XF,Y(J),Z(K)),  
C J=1,2,...,M+1, K=1,2,...,N+1.  
C  
C WHEN LBDCND HAS ANY OTHER VALUE, BDXF IS  
C A DUMMY VARIABLE. BDXF MUST BE  
C DIMENSIONED AT LEAST (M+1)\*(N+1).  
C  
C YS,YF  
C THE RANGE OF Y, I.E. YS .LE. Y .LE. YF.  
C YS MUST BE LESS THAN YF.  
C  
C M  
C 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  
C Y = YS AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Y IS  
C SPECIFIED AT Y = YF.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = YS AND Y = YF.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C AT Y = YS AND THE SOLUTION IS  
C SPECIFIED AT Y=YF.  
C  
C BDYS  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Y AT Y = YS.  
C  
C WHEN MBDCND = 3 OR 4,  
C  
C BDYS(I,K) = (D/DY)U(X(I),YS,Z(K)),  
C I=1,2,...,L+1, K=1,2,...,N+1.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDYS  
C IS A DUMMY VARIABLE. BDYS MUST BE  
C DIMENSIONED AT LEAST (L+1)\*(N+1).  
C  
C BDYF  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Y AT Y = YF.  
C  
C WHEN MBDCND = 2 OR 3,  
C  
C BDYF(I,K) = (D/DY)U(X(I),YF,Z(K)),  
C I=1,2,...,L+1, K=1,2,...,N+1.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDYF  
C IS A DUMMY VARIABLE. BDYF MUST BE  
C DIMENSIONED AT LEAST (L+1)\*(N+1).  
C  
C ZS,ZF  
C THE RANGE OF Z, I.E. ZS .LE. Z .LE. ZF.  
C ZS MUST BE LESS THAN ZF.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (ZS,ZF) IS SUBDIVIDED.  
C HENCE, THERE WILL BE N+1 GRID POINTS  
C IN THE Z-DIRECTION GIVEN BY  
C Z(K) = ZS+(K-1)DZ FOR K=1,2,...,N+1,  
C WHERE DZ = (ZF-ZS)/N IS THE PANEL WIDTH.  
C N MUST BE AT LEAST 5.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT Z = ZS AND Z = ZF.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN Z, I.E.  
C U(I,J,N+K) = U(I,J,K).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C Z = ZS AND Z = ZF.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C Z = ZS AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Z IS  
C SPECIFIED AT Z = ZF.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Z IS SPECIFIED AT  
C Z = ZS AND Z = ZF.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Z IS SPECIFIED AT  
C Z = ZS AND THE SOLUTION IS SPECIFIED  
C AT Z=ZF.  
C  
C BDZS  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Z AT Z = ZS.  
C  
C WHEN NBDCND = 3 OR 4,  
C  
C BDZS(I,J) = (D/DZ)U(X(I),Y(J),ZS),  
C I=1,2,...,L+1, J=1,2,...,M+1.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDZS  
C IS A DUMMY VARIABLE. BDZS MUST BE  
C DIMENSIONED AT LEAST (L+1)\*(M+1).  
C  
C BDZF  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES  
C THE VALUES OF THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Z AT Z = ZF.  
C  
C WHEN NBDCND = 2 OR 3,  
C  
C BDZF(I,J) = (D/DZ)U(X(I),Y(J),ZF),  
C I=1,2,...,L+1, J=1,2,...,M+1.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDZF  
C IS A DUMMY VARIABLE. BDZF MUST BE  
C DIMENSIONED AT LEAST (L+1)\*(M+1).  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .GT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HW3CRT WILL  
C ATTEMPT TO FIND A SOLUTION.  
C  
C LDIMF  
C THE ROW (OR FIRST) DIMENSION OF THE  
C ARRAYS F,BDYS,BDYF,BDZS,AND BDZF AS IT  
C APPEARS IN THE PROGRAM CALLING HW3CRT.  
C THIS PARAMETER IS USED TO SPECIFY THE  
C VARIABLE DIMENSION OF THESE ARRAYS.  
C LDIMF MUST BE AT LEAST L+1.  
C  
C MDIMF  
C THE COLUMN (OR SECOND) DIMENSION OF THE  
C ARRAY F AND THE ROW (OR FIRST) DIMENSION  
C OF THE ARRAYS BDXS AND BDXF AS IT APPEARS  
C IN THE PROGRAM CALLING HW3CRT. THIS  
C PARAMETER IS USED TO SPECIFY THE VARIABLE  
C DIMENSION OF THESE ARRAYS.  
C MDIMF MUST BE AT LEAST M+1.  
C  
C F  
C A THREE-DIMENSIONAL ARRAY OF DIMENSION AT  
C AT LEAST (L+1)\*(M+1)\*(N+1), SPECIFYING THE  
C VALUES OF THE RIGHT SIDE OF THE HELMHOLZ  
C EQUATION AND BOUNDARY VALUES (IF ANY).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I=2,3,...,L, J=2,3,...,M,  
C AND K=2,3,...,N  
C F(I,J,K) = F(X(I),Y(J),Z(K)).  
C  
C ON THE BOUNDARIES, F IS DEFINED AS FOLLOWS:  
C FOR J=1,2,...,M+1, K=1,2,...,N+1,  
C AND I=1,2,...,L+1  
C  
C LBDCND F(1,J,K) F(L+1,J,K)  
C ------ --------------- ---------------  
C  
C 0 F(XS,Y(J),Z(K)) F(XS,Y(J),Z(K))  
C 1 U(XS,Y(J),Z(K)) U(XF,Y(J),Z(K))  
C 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 MBDCND F(I,1,K) F(I,M+1,K)  
C ------ --------------- ---------------  
C  
C 0 F(X(I),YS,Z(K)) F(X(I),YS,Z(K))  
C 1 U(X(I),YS,Z(K)) U(X(I),YF,Z(K))  
C 2 U(X(I),YS,Z(K)) F(X(I),YF,Z(K))  
C 3 F(X(I),YS,Z(K)) F(X(I),YF,Z(K))  
C 4 F(X(I),YS,Z(K)) U(X(I),YF,Z(K))  
C  
C NBDCND F(I,J,1) F(I,J,N+1)  
C ------ --------------- ---------------  
C  
C 0 F(X(I),Y(J),ZS) F(X(I),Y(J),ZS)  
C 1 U(X(I),Y(J),ZS) U(X(I),Y(J),ZF)  
C 2 U(X(I),Y(J),ZS) F(X(I),Y(J),ZF)  
C 3 F(X(I),Y(J),ZS) F(X(I),Y(J),ZF)  
C 4 F(X(I),Y(J),ZS) U(X(I),Y(J),ZF)  
C  
C NOTE:  
C IF THE TABLE CALLS FOR BOTH THE SOLUTION  
C U AND THE RIGHT SIDE F ON A BOUNDARY,  
C THEN THE SOLUTION MUST BE SPECIFIED.  
C  
C  
C ON OUTPUT F  
C CONTAINS THE SOLUTION U(I,J,K) OF THE  
C FINITE DIFFERENCE APPROXIMATION FOR THE  
C GRID POINT (X(I),Y(J),Z(K)) FOR  
C I=1,2,...,L+1, J=1,2,...,M+1,  
C AND K=1,2,...,N+1.  
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. PWSCRT  
C THEN COMPUTES THIS SOLUTION, WHICH IS A  
C LEAST SQUARES SOLUTION TO THE ORIGINAL  
C 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, A SOLUTION  
C IS 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 INVALID INPUT  
C PARAMETERS. EXCEPT 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  
C = 3 LBDCND .LT. 0 .OR. LBDCND .GT. 4  
C = 4 YS .GE. YF  
C = 5 M .LT. 5  
C = 6 MBDCND .LT. 0 .OR. MBDCND .GT. 4  
C = 7 ZS .GE. ZF  
C = 8 N .LT. 5  
C = 9 NBDCND .LT. 0 .OR. NBDCND .GT. 4  
C = 10 LDIMF .LT. L+1  
C = 11 MDIMF .LT. M+1  
C = 12 LAMBDA .GT. 0  
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 SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HW3CRT, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
C SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED Files fish.f,pois3d.f,fftpack.f,comf.f  
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.  
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 BOUNDARY DATA, AND  
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.  
C MORE DETAILED INFORMATION ABOUT ACCURACY  
C CAN BE FOUND IN THE DOCUMENTATION FOR  
C ROUTINE POIS3D WHICH IS THE ROUTINE THAT  
C ACTUALLY SOLVES THE FINITE DIFFERENCE  
C EQUATIONS.  
C  
C REFERENCES NONE  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HWSCRT**

C  
C file hwscrt.txt (documentation for the FISHPACK solver HWSCRT)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HWSCRT (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C DIMENSION OF BDA(N), BDB(N), BDC(M),BDD(M),  
C ARGUMENTS F(IDIMF,N)  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE STANDARD FIVE-POINT FINITE  
C DIFFERENCE APPROXIMATION TO THE HELMHOLTZ  
C EQUATION IN CARTESIAN COORDINATES. THIS  
C EQUATION IS  
C  
C (D/DX)(DU/DX) + (D/DY)(DU/DY)  
C + LAMBDA\*U = F(X,Y).  
C  
C USAGE CALL HWSCRT (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT A,B  
C  
C THE RANGE OF X, I.E., A .LE. X .LE. B.  
C A MUST BE LESS THAN B.  
C  
C M  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (A,B) IS SUBDIVIDED.  
C HENCE, THERE WILL BE M+1 GRID POINTS  
C IN THE X-DIRECTION GIVEN BY  
C X(I) = A+(I-1)DX FOR I = 1,2,...,M+1,  
C WHERE DX = (B-A)/M IS THE PANEL WIDTH.  
C M MUST BE GREATER THAN 3.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT X = A AND X = B.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN X,  
C I.E., U(I,J) = U(M+I,J).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C X = A AND X = B.  
C = 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  
C AT X = A AND X = B.  
C = 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.  
C  
C BDA  
C 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.  
C  
C WHEN MBDCND = 3 OR 4,  
C  
C 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  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1  
C THAT SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO X AT X = B.  
C  
C WHEN MBDCND = 2 OR 3,  
C  
C BDB(J) = (D/DX)U(B,Y(J)), J = 1,2,...,N+1  
C  
C WHEN MBDCND HAS ANY OTHER VALUE BDB IS A  
C DUMMY VARIABLE.  
C  
C C,D  
C THE RANGE OF Y, I.E., C .LE. Y .LE. D.  
C C MUST BE LESS THAN D.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (C,D) IS SUBDIVIDED. HENCE,  
C THERE WILL BE N+1 GRID POINTS IN THE  
C Y-DIRECTION GIVEN BY Y(J) = C+(J-1)DY  
C FOR J = 1,2,...,N+1, WHERE  
C DY = (D-C)/N IS THE PANEL WIDTH.  
C N MUST BE GREATER THAN 3.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS AT  
C Y = C AND Y = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN Y,  
C I.E., U(I,J) = U(I,N+J).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C Y = C AND Y = D.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C Y = C AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO Y IS  
C SPECIFIED AT Y = D.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = C AND Y = D.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Y IS SPECIFIED AT  
C Y = C AND THE SOLUTION IS SPECIFIED  
C AT Y = D.  
C  
C BDC  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO Y AT Y = C.  
C  
C WHEN NBDCND = 3 OR 4,  
C  
C BDC(I) = (D/DY)U(X(I),C), I = 1,2,...,M+1  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC IS  
C A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO Y AT Y = D.  
C  
C WHEN NBDCND = 2 OR 3,  
C  
C BDD(I) = (D/DY)U(X(I),D), I = 1,2,...,M+1  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDD IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .GT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HWSCRT WILL  
C ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY, OF DIMENSION AT  
C LEAST (M+1)\*(N+1), SPECIFYING VALUES OF THE  
C RIGHT SIDE OF THE HELMHOLTZ EQUATION AND  
C BOUNDARY VALUES (IF ANY).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I = 2,3,...,M AND J = 2,3,...,N  
C F(I,J) = F(X(I),Y(J)).  
C  
C ON THE BOUNDARIES, F IS DEFINED AS FOLLOWS:  
C FOR J=1,2,...,N+1, I=1,2,...,M+1,  
C  
C MBDCND F(1,J) F(M+1,J)  
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))  
C 4 F(A,Y(J)) U(B,Y(J))  
C  
C  
C NBDCND F(I,1) F(I,N+1)  
C ------ --------- --------  
C  
C 0 F(X(I),C) F(X(I),C)  
C 1 U(X(I),C) U(X(I),D)  
C 2 U(X(I),C) F(X(I),D)  
C 3 F(X(I),C) F(X(I),D)  
C 4 F(X(I),C) U(X(I),D)  
C  
C NOTE:  
C IF THE TABLE CALLS FOR BOTH THE SOLUTION U  
C AND THE RIGHT SIDE F AT A CORNER THEN THE  
C 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  
C HWSCRT. 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 DIFFERENCE APPROXIMATION FOR THE GRID POINT  
C (X(I),Y(J)), I = 1,2,...,M+1,  
C J = 1,2,...,N+1 .  
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 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 INVALID INPUT  
C PARAMETERS. EXCEPT FOR NUMBERS 0 AND 6,  
C A SOLUTION IS NOT ATTEMPTED.  
C  
C = 0 NO ERROR  
C = 1 A .GE. B  
C = 2 MBDCND .LT. 0 OR MBDCND .GT. 4  
C = 3 C .GE. D  
C = 4 N .LE. 3  
C = 5 NBDCND .LT. 0 OR NBDCND .GT. 4  
C = 6 LAMBDA .GT. 0  
C = 7 IDIMF .LT. M+1  
C = 8 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  
C SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSIBLY INCORRECT CALL TO HWSCRT, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
C  
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  
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.  
c Revised in June 2004 by John Adams using  
c Fortran 90 dynamically allocated work space.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, AND  
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)  
C BUT ALSO DEPENDS ON INPUT PARAMETERS NBDCND  
C AND MBDCND.  
C  
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.  
C  
C REFERENCES SWARZTRAUBER,P. AND R. SWEET, "EFFICIENT  
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF  
C ELLIPTIC EQUATIONS"  
C NCAR TN/IA-109, JULY, 1975, 138 PP.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HWSCSP**

C  
C file hwscsp.txt (documentation for the FISHPACK solver HWSCSP)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HWSCSP (INTL,TS,TF,M,MBDCND,BDTS,BDTF,RS,RF,N,NBDCND,  
C + BDRS,BDRF,ELMBDA,F,IDIMF,PERTRB,IERROR,W)  
C  
C  
C DIMENSION OF BDTS(N+1), BDTF(N+1), BDRS(M+1), BDRF(M+1),  
C ARGUMENTS F(IDIMF,N+1)  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES A FINITE DIFFERENCE APPROXIMATION  
C TO THE MODIFIED HELMHOLTZ EQUATION IN  
C SPHERICAL COORDINATES ASSUMING AXISYMMETRY  
C (NO DEPENDENCE ON LONGITUDE). THE EQUATION  
C IS  
C  
C (1/R\*\*2)(D/DR)((R\*\*2)(D/DR)U) +  
C  
C (1/(R\*\*2)SIN(THETA))(D/DTHETA)  
C  
C (SIN(THETA)(D/DTHETA)U) +  
C  
C (LAMBDA/(RSIN(THETA))\*\*2)U = F(THETA,R).  
C  
C THIS TWO DIMENSIONAL MODIFIED HELMHOLTZ  
C EQUATION RESULTS FROM THE FOURIER TRANSFORM  
C OF THE THREE DIMENSIONAL POISSON EQUATION.  
C  
C USAGE CALL HWSCSP (INTL,TS,TF,M,MBDCND,BDTS,BDTF,  
C RS,RF,N,NBDCND,BDRS,BDRF,ELMBDA,  
C 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 HWSCSP.  
C  
C 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  
C HAS BEEN MADE THEN SUBSEQUENT SOLUTIONS  
C CORRESPONDING TO DIFFERENT F, BDTS, BDTF,  
C BDRS, BDRF CAN BE OBTAINED FASTER WITH  
C INTL = 1 SINCE INITIALIZATION IS NOT  
C REPEATED.  
C  
C TS,TF  
C THE RANGE OF THETA (COLATITUDE), I.E.,  
C 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 SOUTH POLE.  
C  
C \*\*\*\* IMPORTANT \*\*\*\*  
C  
C IF TF IS EQUAL TO PI THEN IT MUST BE  
C COMPUTED USING THE STATEMENT  
C TF = PIMACH(DUM). THIS INSURES THAT TF  
C IN THE USER'S PROGRAM IS EQUAL TO PI IN  
C 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  
C INTERVAL (TS,TF) IS SUBDIVIDED.  
C HENCE, THERE WILL BE M+1 GRID POINTS  
C IN THE THETA-DIRECTION GIVEN BY  
C THETA(K) = (I-1)DTHETA+TS FOR  
C I = 1,2,...,M+1, WHERE DTHETA = (TF-TS)/M  
C IS THE PANEL WIDTH.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITION  
C AT THETA = TS AND THETA = TF.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = TS AND THETA = TF.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C THETA = TS AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = TF  
C (SEE NOTE 2 BELOW).  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = TS AND THETA = TF  
C (SEE NOTES 1,2 BELOW).  
C = 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 = 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  
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  
C THETA = TS = 0 AND THETA = TF = PI.  
C  
C NOTE 1:  
C IF TS = 0, DO NOT USE MBDCND = 3,4, OR 8,  
C BUT INSTEAD USE MBDCND = 5,6, OR 9 .  
C  
C NOTE 2:  
C IF TF = PI, DO NOT USE MBDCND = 2,3, OR 6,  
C BUT INSTEAD USE MBDCND = 7,8, OR 9 .  
C  
C BDTS  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO THETA AT  
C THETA = TS. WHEN MBDCND = 3,4, OR 8,  
C  
C BDTS(J) = (D/DTHETA)U(TS,R(J)),  
C J = 1,2,...,N+1 .  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDTS IS  
C A DUMMY VARIABLE.  
C  
C BDTF  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO THETA AT  
C THETA = TF. WHEN MBDCND = 2,3, OR 6,  
C  
C BDTF(J) = (D/DTHETA)U(TF,R(J)),  
C J = 1,2,...,N+1 .  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDTF IS  
C A DUMMY VARIABLE.  
C  
C RS,RF  
C THE RANGE OF R, I.E., RS .LE. R .LT. RF.  
C RS MUST BE LESS THAN RF. RS MUST BE  
C NON-NEGATIVE.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (RS,RF) IS SUBDIVIDED.  
C HENCE, THERE WILL BE N+1 GRID POINTS IN THE  
C R-DIRECTION GIVEN BY R(J) = (J-1)DR+RS  
C FOR J = 1,2,...,N+1, WHERE DR = (RF-RS)/N  
C IS THE PANEL WIDTH.  
C N MUST BE GREATER THAN 2  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITION  
C AT R = RS AND R = RF.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C R = RS AND R = RF.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C R = RS AND THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO R  
C IS SPECIFIED AT R = RF.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C 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.  
C = 6 IF THE SOLUTION IS UNSPECIFIED AT  
C R = RS = 0 (SEE NOTE BELOW) AND THE  
C DERIVATIVE OF THE SOLUTION WITH  
C RESPECT TO R IS SPECIFIED AT R = RF.  
C  
C NOTE:  
C NBDCND = 5 OR 6 CANNOT BE USED WITH  
C MBDCND = 1,2,4,5, OR 7. THE FORMER  
C INDICATES THAT THE SOLUTION IS UNSPECIFIED  
C AT R = 0, THE LATTER INDICATES THAT THE  
C SOLUTION IS SPECIFIED).  
C USE INSTEAD NBDCND = 1 OR 2 .  
C  
C BDRS  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO R AT R = RS.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDRS(I) = (D/DR)U(THETA(I),RS),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDRS IS  
C A DUMMY VARIABLE.  
C  
C BDRF  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1  
C THAT SPECIFIES THE VALUES OF THE  
C DERIVATIVE OF THE SOLUTION WITH RESPECT  
C TO R AT R = RF.  
C  
C WHEN NBDCND = 2,3, OR 6,  
C BDRF(I) = (D/DR)U(THETA(I),RF),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDRF IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .GT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HWSCSP WILL  
C ATTEMPT TO FIND A SOLUTION. IF NBDCND = 5  
C OR 6 OR MBDCND = 5,6,7,8, OR 9, ELMBDA  
C MUST BE ZERO.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY, OF DIMENSION AT  
C LEAST (M+1)\*(N+1), SPECIFYING VALUES OF THE  
C RIGHT SIDE OF THE HELMHOLTZ EQUATION AND  
C BOUNDARY VALUES (IF ANY).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I = 2,3,...,M AND J = 2,3,...,N  
C F(I,J) = F(THETA(I),R(J)).  
C  
C ON THE BOUNDARIES, F IS DEFINED AS FOLLOWS:  
C FOR J=1,2,...,N+1, I=1,2,...,M+1,  
C  
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 3 F(TS,R(J)) F(TF,R(J))  
C 4 F(TS,R(J)) U(TF,R(J))  
C 5 F(0,R(J)) U(TF,R(J))  
C 6 F(0,R(J)) F(TF,R(J))  
C 7 U(TS,R(J)) F(PI,R(J))  
C 8 F(TS,R(J)) F(PI,R(J))  
C 9 F(0,R(J)) F(PI,R(J))  
C  
C NBDCND F(I,1) F(I,N+1)  
C ------ -------------- --------------  
C  
C 1 U(THETA(I),RS) U(THETA(I),RF)  
C 2 U(THETA(I),RS) F(THETA(I),RF)  
C 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 IF THE TABLE CALLS FOR BOTH THE SOLUTION  
C U AND THE RIGHT SIDE F AT A CORNER THEN  
C 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  
C HWSCSP. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F. IDIMF MUST  
C BE AT LEAST 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 calling HWSCSP must be:  
c  
c USE fish  
c  
c 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  
c 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  
c include the statement:  
c  
c CALL FISHFIN(W)  
C  
C after the final approximation is generated by  
C HWSCSP. The will deallocate the real and complex  
c work space of W. Failure to include this statement  
c could result in serious memory leakage.  
c  
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),R(J)), I = 1,2,...,M+1,  
C J = 1,2,...,N+1 .  
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. HWSCSP  
C THEN COMPUTES THIS SOLUTION, WHICH IS A  
C LEAST SQUARES SOLUTION TO THE ORIGINAL  
C APPROXIMATION. THIS SOLUTION IS NOT UNIQUE  
C AND IS UNNORMALIZED. THE VALUE OF PERTRB  
C SHOULD BE SMALL COMPARED TO THE RIGHT SIDE  
C F. OTHERWISE , A SOLUTION IS OBTAINED TO  
C AN ESSENTIALLY DIFFERENT PROBLEM. THIS  
C COMPARISON SHOULD ALWAYS BE MADE TO INSURE  
C THAT A MEANINGFUL SOLUTION HAS BEEN OBTAINED.  
C  
C IERROR  
C AN ERROR FLAG THAT INDICATES INVALID INPUT  
C PARAMETERS. EXCEPT FOR NUMBERS 0 AND 10,  
C A SOLUTION IS NOT ATTEMPTED.  
C  
C = 1 TS.LT.0. OR TF.GT.PI  
C = 2 TS.GE.TF  
C = 3 M.LT.5  
C = 4 MBDCND.LT.1 OR MBDCND.GT.9  
C = 5 RS.LT.0  
C = 6 RS.GE.RF  
C = 7 N.LT.5  
C = 8 NBDCND.LT.1 OR NBDCND.GT.6  
C = 9 ELMBDA.GT.0  
C = 10 IDIMF.LT.M+1  
C = 11 ELMBDA.NE.0 AND MBDCND.GE.5  
C = 12 ELMBDA.NE.0 AND NBDCND EQUALS 5 OR 6  
C = 13 MBDCND EQUALS 5,6 OR 9 AND TS.NE.0  
C = 14 MBDCND.GE.7 AND TF.NE.PI  
C = 15 TS.EQ.0 AND MBDCND EQUALS 3,4 OR 8  
C = 16 TF.EQ.PI AND MBDCND EQUALS 2,3 OR 6  
C = 17 NBDCND.GE.5 AND RS.NE.0  
C = 18 NBDCND.GE.5 AND MBDCND EQUALS 1,2,4,5 OR  
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  
C SINCE THIS IS THE ONLY MEANS OF INDICATING  
C A POSSLIBY INCORRECT CALL TO HWSCSP, THE  
C USER SHOULD TEST IERROR AFTER A CALL.  
C  
C W  
c The derived type (fishworkspace) variable W  
c contains real and complex values that must not  
C be destroyed if HWSCSP is called again with  
C INTL=1.  
C  
C SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED files fish.f,blktri.f,comf.f  
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. Revised by John  
c Adams in June 2004 using Fortran 90 dynamically  
C allocated work space and derived datat types  
c to eliminate mixed mode conflicts in the earlier  
c versions.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, AND  
C ADJUSTS THE RIGHT SIDE OF SINGULAR SYSTEMS  
C AND THEN CALLS BLKTRI TO SOLVE THE SYSTEM.  
C  
C REFERENCES SWARZTRAUBER,P. AND R. SWEET, "EFFICIENT  
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF  
C ELLIPTIC EQUATIONS"  
C NCAR TN/IA-109, JULY, 1975, 138 PP.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HWSCYL**

C  
C file hwscyl.txt (documentation for the FISHPACK solver HWSCYL)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HWSCYL (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N+1)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES A FINITE DIFFERENCE APPROXIMATION  
C TO THE HELMHOLTZ EQUATION IN CYLINDRICAL  
C COORDINATES. THIS MODIFIED HELMHOLTZ EQUATION  
C  
C (1/R)(D/DR)(R(DU/DR)) + (D/DZ)(DU/DZ)  
C  
C + (LAMBDA/R\*\*2)U = F(R,Z)  
C  
C RESULTS FROM THE FOURIER TRANSFORM OF THE  
C THREE-DIMENSIONAL POISSON EQUATION.  
C  
C USAGE CALL HWSCYL (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR,W)  
C  
C ARGUMENTS  
C ON INPUT A,B  
C THE RANGE OF R, I.E., A .LE. R .LE. B.  
C A MUST BE LESS THAN B AND A MUST BE  
C NON-NEGATIVE.  
C  
C M  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (A,B) IS SUBDIVIDED. HENCE,  
C THERE WILL BE M+1 GRID POINTS IN THE  
C R-DIRECTION GIVEN BY R(I) = A+(I-1)DR,  
C FOR I = 1,2,...,M+1, WHERE DR = (B-A)/M  
C IS THE PANEL WIDTH. M MUST BE GREATER  
C THAN 3.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT R = A AND R = B.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C R = A AND R = B.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C R = A AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO R IS  
C SPECIFIED AT R = B.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND R = B.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND THE  
C SOLUTION IS SPECIFIED AT R = B.  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C R = A = 0 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  
C AT R = B.  
C  
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  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO R AT R = A.  
C  
C WHEN MBDCND = 3 OR 4,  
C BDA(J) = (D/DR)U(A,Z(J)), J = 1,2,...,N+1.  
C  
C 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  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO R AT R = B.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DR)U(B,Z(J)), J = 1,2,...,N+1.  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDB IS  
C A DUMMY VARIABLE.  
C  
C C,D  
C THE RANGE OF Z, I.E., C .LE. Z .LE. D.  
C C MUST BE LESS THAN D.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (C,D) IS SUBDIVIDED. HENCE,  
C THERE WILL BE N+1 GRID POINTS IN THE  
C Z-DIRECTION GIVEN BY Z(J) = C+(J-1)DZ,  
C FOR J = 1,2,...,N+1,  
C WHERE DZ = (D-C)/N IS THE PANEL WIDTH.  
C N MUST BE GREATER THAN 3.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT Z = C AND Z = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN Z,  
C I.E., U(I,1) = U(I,N+1).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C Z = C AND Z = D.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C Z = C AND THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO Z IS  
C SPECIFIED AT Z = D.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO Z IS  
C SPECIFIED AT Z = C AND Z = D.  
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.  
C  
C BDC  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO Z AT Z = C.  
C  
C WHEN NBDCND = 3 OR 4,  
C BDC(I) = (D/DZ)U(R(I),C), I = 1,2,...,M+1.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC IS  
C A DUMMY VARIABLE.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO Z AT Z = D.  
C  
C WHEN NBDCND = 2 OR 3,  
C BDD(I) = (D/DZ)U(R(I),D), I = 1,2,...,M+1  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDD IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .GT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HWSCYL WILL  
C ATTEMPT TO FIND A SOLUTION. LAMBDA MUST  
C BE ZERO WHEN MBDCND = 5 OR 6 .  
C  
C F  
C A TWO-DIMENSIONAL ARRAY, OF DIMENSION AT  
C LEAST (M+1)\*(N+1), SPECIFYING VALUES  
C OF THE RIGHT SIDE OF THE HELMHOLTZ  
C EQUATION AND BOUNDARY DATA (IF ANY).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I = 2,3,...,M AND J = 2,3,...,N  
C F(I,J) = F(R(I),Z(J)).  
C  
C ON THE 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 ------ --------- ---------  
C  
C 1 U(A,Z(J)) U(B,Z(J))  
C 2 U(A,Z(J)) F(B,Z(J))  
C 3 F(A,Z(J)) F(B,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))  
C  
C NBDCND F(I,1) F(I,N+1)  
C ------ --------- ---------  
C  
C 0 F(R(I),C) F(R(I),C)  
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)  
C 4 F(R(I),C) U(R(I),D)  
C  
C NOTE:  
C IF THE TABLE CALLS FOR BOTH THE SOLUTION  
C U AND THE RIGHT SIDE F AT A CORNER THEN  
C 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  
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 DIFFERENCE APPROXIMATION FOR THE GRID POINT  
C (R(I),Z(J)), I =1,2,...,M+1, J =1,2,...,N+1.  
C  
C PERTRB  
C IF ONE SPECIFIES A COMBINATION OF PERIODIC,  
C DERIVATIVE, AND 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. 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.  
C 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  
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  
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  
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 .  
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  
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 HWSCYL, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
C  
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  
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.  
c Revised in June 2004 by John Adams using  
c Fortran 90 dynamically allocated work space.  
C  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, AND  
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)  
C BUT ALSO DEPENDS ON INPUT PARAMETERS NBDCND  
C AND MBDCND.  
C  
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.  
C  
C REFERENCES SWARZTRAUBER,P. AND R. SWEET, "EFFICIENT  
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF  
C ELLIPTIC EQUATIONS"  
C NCAR TN/IA-109, JULY, 1975, 138 PP.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HWSPLR**

C  
C file hwsplr.txt (documentation for the FISHPACK solver HWSPLR)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HWSPLR (A,B,M,MBDCND,BDA,BDB,C,D,N,NBDCND,BDC,BDD,  
C + ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C  
C DIMENSION OF BDA(N),BDB(N),BDC(M),BDD(M),F(IDIMF,N+1)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES A FINITE DIFFERENCE APPROXIMATION TO  
C THE HELMHOLTZ EQUATION IN POLAR COORDINATES.  
C THE EQUATION IS  
C  
C (1/R)(D/DR)(R(DU/DR)) +  
C (1/R\*\*2)(D/DTHETA)(DU/DTHETA) +  
C LAMBDA\*U = F(R,THETA).  
C  
C USAGE CALL HWSPLR (A,B,M,MBDCND,BDA,BDB,C,D,N,  
C NBDCND,BDC,BDD,ELMBDA,F,IDIMF,  
C PERTRB,IERROR,W)  
C  
C ARGUMENTS  
C ON INPUT A,B  
C THE RANGE OF R, I.E., A .LE. R .LE. B.  
C A MUST BE LESS THAN B AND A MUST BE  
C NON-NEGATIVE.  
C  
C M  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (A,B) IS SUBDIVIDED. HENCE,  
C THERE WILL BE M+1 GRID POINTS IN THE  
C R-DIRECTION GIVEN BY R(I) = A+(I-1)DR,  
C FOR I = 1,2,...,M+1,  
C WHERE DR = (B-A)/M IS THE PANEL WIDTH.  
C M MUST BE GREATER THAN 3.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITION  
C AT R = A AND R = B.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C R = A AND R = B.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C R = A AND THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO R IS  
C SPECIFIED AT R = B.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND R = B.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO R IS SPECIFIED AT  
C R = A (SEE NOTE BELOW) AND THE  
C SOLUTION IS SPECIFIED AT R = B.  
C = 5 IF THE SOLUTION IS UNSPECIFIED AT  
C R = A = 0 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  
C AT R = B.  
C  
C NOTE:  
C IF A = 0, DO NOT USE MBDCND = 3 OR 4, BUT  
C INSTEAD USE MBDCND = 1,2,5, OR 6 .  
C  
C BDA  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO R AT R = A.  
C  
C WHEN MBDCND = 3 OR 4,  
C BDA(J) = (D/DR)U(A,THETA(J)),  
C J = 1,2,...,N+1 .  
C  
C 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  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO R AT R = B.  
C  
C WHEN MBDCND = 2,3, OR 6,  
C BDB(J) = (D/DR)U(B,THETA(J)),  
C J = 1,2,...,N+1 .  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDB IS  
C A DUMMY VARIABLE.  
C  
C C,D  
C THE RANGE OF THETA, I.E., C .LE.  
C THETA .LE. D. C MUST BE LESS THAN D.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (C,D) IS SUBDIVIDED. HENCE,  
C THERE WILL BE N+1 GRID POINTS IN THE  
C THETA-DIRECTION GIVEN BY  
C THETA(J) = C+(J-1)DTHETA FOR  
C J = 1,2,...,N+1, WHERE  
C DTHETA = (D-C)/N IS THE PANEL WIDTH.  
C N MUST BE GREATER THAN 3.  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITIONS  
C AT THETA = C AND AT THETA = D.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN THETA,  
C I.E., U(I,J) = U(I,N+J).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = C AND THETA = D  
C (SEE NOTE BELOW).  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C THETA = C AND THE DERIVATIVE OF THE  
C SOLUTION WITH RESPECT TO THETA IS  
C SPECIFIED AT THETA = D  
C (SEE NOTE BELOW).  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO THETA IS SPECIFIED  
C AT THETA = C AND THE SOLUTION IS  
C SPECIFIED AT THETA = D  
C (SEE NOTE BELOW).  
C  
C NOTE:  
C WHEN NBDCND = 1,2, OR 4, DO NOT USE  
C MBDCND = 5 OR 6  
C (THE FORMER INDICATES THAT THE SOLUTION  
C IS SPECIFIED AT R = 0, THE LATTER INDICATES  
C THE SOLUTION IS UNSPECIFIED AT R = 0).  
C USE INSTEAD MBDCND = 1 OR 2 .  
C  
C BDC  
C 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 THETA = C. WHEN NBDCND = 3 OR 4,  
C  
C BDC(I) = (D/DTHETA)U(R(I),C),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC IS  
C A DUMMY VARIABLE.  
C  
C BDD  
C 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 THETA = D. WHEN NBDCND = 2 OR 3,  
C  
C BDD(I) = (D/DTHETA)U(R(I),D),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDD IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .LT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HWSPLR WILL  
C ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY, OF DIMENSION AT  
C LEAST (M+1)\*(N+1), SPECIFYING VALUES  
C OF THE RIGHT SIDE OF THE HELMHOLTZ  
C EQUATION AND BOUNDARY DATA (IF ANY).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I = 2,3,...,M AND J = 2,3,...,N  
C F(I,J) = F(R(I),THETA(J)).  
C  
C ON THE 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 ------ ------------- -------------  
C  
C 1 U(A,THETA(J)) U(B,THETA(J))  
C 2 U(A,THETA(J)) F(B,THETA(J))  
C 3 F(A,THETA(J)) F(B,THETA(J))  
C 4 F(A,THETA(J)) U(B,THETA(J))  
C 5 F(0,0) U(B,THETA(J))  
C 6 F(0,0) F(B,THETA(J))  
C  
C NBDCND F(I,1) F(I,N+1)  
C ------ --------- ---------  
C  
C 0 F(R(I),C) F(R(I),C)  
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)  
C 4 F(R(I),C) U(R(I),D)  
C  
C NOTE:  
C 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  
C HWSPLR. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F. IDIMF MUST  
C BE AT LEAST M+1.  
C  
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  
C SOLUTION EXISTS. HWSPLR 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. PERTRB SHOULD BE SMALL COMPARED  
C TO THE RIGHT SIDE. OTHERWISE, A SOLUTION  
C IS 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 INVALID INPUT  
C PARAMETERS. EXCEPT FOR NUMBERS 0 AND 11,  
C 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  
C = 6 NBDCND .LT. 0 OR .GT. 4 .  
C = 7 A = 0, MBDCND = 3 OR 4 .  
C = 8 A .GT. 0, MBDCND .GE. 5 .  
C = 9 MBDCND .GE. 5, NBDCND .NE. 0  
C AND NBDCND .NE. 3 .  
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  
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 HWSPLR, THE  
C USER SHOULD TEST IERROR AFTER THE CALL.  
C  
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  
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.  
c Revised in June 2004 by John Adams using  
c Fortran 90 dynamically allocated work space.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, AND  
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)  
C BUT ALSO DEPENDS ON INPUT PARAMETERS NBDCND  
C AND MBDCND.  
C  
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.  
C  
C REFERENCES SWARZTRAUBER,P. AND R. SWEET, "EFFICIENT  
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF  
C ELLIPTIC EQUATIONS"  
C NCAR TN/IA-109, JULY, 1975, 138 PP.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**HWSSSP**

C  
C file hwsssp.txt (documentation for the FISHPACK solver HWSSSP)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE HWSSSP (TS,TF,M,MBDCND,BDTS,BDTF,PS,PF,N,NBDCND,BDPS,  
C + BDPF,ELMBDA,F,IDIMF,PERTRB,IERROR)  
C  
C DIMENSION OF BDTS(N+1), BDTF(N+1), BDPS(M+1), BDPF(M+1),  
C ARGUMENTS F(IDIMF,N+1)  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES A FINITE DIFFERENCE APPROXIMATION TO  
C THE HELMHOLTZ EQUATION IN SPHERICAL  
C COORDINATES AND ON THE SURFACE OF THE UNIT  
C SPHERE (RADIUS OF 1). THE EQUATION IS  
C  
C (1/SIN(THETA))(D/DTHETA)(SIN(THETA)  
C (DU/DTHETA)) + (1/SIN(THETA)\*\*2)(D/DPHI)  
C (DU/DPHI) + LAMBDA\*U = F(THETA,PHI)  
C  
C WHERE THETA IS COLATITUDE AND PHI IS  
C LONGITUDE.  
C  
C USAGE CALL HWSSSP (TS,TF,M,MBDCND,BDTS,BDTF,PS,PF,  
C N,NBDCND,BDPS,BDPF,ELMBDA,F,  
C IDIMF,PERTRB,IERROR,W)  
C  
C ARGUMENTS  
C ON INPUT TS,TF  
C  
C THE RANGE OF THETA (COLATITUDE), I.E.,  
C TS .LE. THETA .LE. TF. TS MUST BE LESS  
C THAN TF. TS AND TF ARE IN RADIANS.  
C A TS OF ZERO CORRESPONDS TO THE NORTH  
C POLE AND A TF OF PI CORRESPONDS TO  
C THE SOUTH POLE.  
C  
C \* \* \* IMPORTANT \* \* \*  
C  
C IF TF IS EQUAL TO PI THEN IT MUST BE  
C COMPUTED USING THE STATEMENT  
C TF = PIMACH(DUM). THIS INSURES THAT TF  
C IN THE USER'S PROGRAM IS EQUAL TO PI IN  
C THIS PROGRAM WHICH PERMITS SEVERAL TESTS  
C OF THE INPUT PARAMETERS THAT OTHERWISE  
C WOULD NOT BE POSSIBLE.  
C  
C  
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,...,M+1, 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  
C AT THETA = TS AND THETA = TF.  
C  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C THETA = TS AND THETA = TF.  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C 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 SOLUTION  
C 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 SPECIFIED  
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  
C IS 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) AND  
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  
C NOTES:  
C IF TS = 0, DO NOT USE MBDCND = 3,4, OR 8,  
C BUT INSTEAD USE MBDCND = 5,6, OR 9 .  
C  
C IF TF = PI, DO NOT USE MBDCND = 2,3, OR 6,  
C BUT INSTEAD USE MBDCND = 7,8, OR 9 .  
C  
C BDTS  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE OF  
C THE SOLUTION WITH RESPECT TO THETA AT  
C THETA = TS. WHEN MBDCND = 3,4, OR 8,  
C  
C BDTS(J) = (D/DTHETA)U(TS,PHI(J)),  
C J = 1,2,...,N+1 .  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDTS IS  
C A DUMMY VARIABLE.  
C  
C BDTF  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1  
C THAT SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO THETA AT  
C THETA = TF. WHEN MBDCND = 2,3, OR 6,  
C  
C BDTF(J) = (D/DTHETA)U(TF,PHI(J)),  
C J = 1,2,...,N+1 .  
C  
C WHEN MBDCND HAS ANY OTHER VALUE, BDTF IS  
C A DUMMY VARIABLE.  
C  
C PS,PF  
C THE RANGE OF PHI (LONGITUDE), I.E.,  
C PS .LE. PHI .LE. PF. PS MUST BE LESS  
C THAN PF. PS AND PF ARE IN RADIANS.  
C IF PS = 0 AND PF = 2\*PI, PERIODIC  
C BOUNDARY CONDITIONS ARE USUALLY PRESCRIBED.  
C  
C \* \* \* IMPORTANT \* \* \*  
C  
C IF PF IS EQUAL TO 2\*PI THEN IT MUST BE  
C COMPUTED USING THE STATEMENT  
C PF = 2.\*PIMACH(DUM). THIS INSURES THAT  
C PF IN THE USERS PROGRAM IS EQUAL TO  
C 2\*PI IN THIS PROGRAM WHICH PERMITS TESTS  
C OF THE INPUT PARAMETERS THAT OTHERWISE  
C WOULD NOT BE POSSIBLE.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (PS,PF) IS SUBDIVIDED.  
C HENCE, THERE WILL BE N+1 GRID POINTS  
C IN THE PHI-DIRECTION GIVEN BY  
C PHI(J) = (J-1)DPHI+PS FOR  
C J = 1,2,...,N+1, WHERE  
C DPHI = (PF-PS)/N IS THE PANEL WIDTH.  
C N MUST BE GREATER THAN 4  
C  
C NBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITION  
C AT PHI = PS AND PHI = PF.  
C  
C = 0 IF THE SOLUTION IS PERIODIC IN PHI,  
C I.U., U(I,J) = U(I,N+J).  
C = 1 IF THE SOLUTION IS SPECIFIED AT  
C PHI = PS AND PHI = PF  
C (SEE NOTE BELOW).  
C = 2 IF THE SOLUTION IS SPECIFIED AT  
C PHI = PS (SEE NOTE BELOW)  
C AND THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO PHI IS SPECIFIED  
C AT PHI = PF.  
C = 3 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO PHI IS SPECIFIED  
C AT PHI = PS AND PHI = PF.  
C = 4 IF THE DERIVATIVE OF THE SOLUTION  
C WITH RESPECT TO PHI IS SPECIFIED  
C AT PS AND THE SOLUTION IS SPECIFIED  
C AT PHI = PF  
C  
C NOTE:  
C NBDCND = 1,2, OR 4 CANNOT BE USED WITH  
C MBDCND = 5,6,7,8, OR 9. THE FORMER INDICATES  
C THAT THE SOLUTION IS SPECIFIED AT A POLE, THE  
C LATTER INDICATES THAT THE SOLUTION IS NOT  
C SPECIFIED. USE INSTEAD MBDCND = 1 OR 2.  
C  
C BDPS  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO PHI AT  
C PHI = PS. WHEN NBDCND = 3 OR 4,  
C  
C BDPS(I) = (D/DPHI)U(THETA(I),PS),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDPS IS  
C A DUMMY VARIABLE.  
C  
C BDPF  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUES OF THE DERIVATIVE  
C OF THE SOLUTION WITH RESPECT TO PHI AT  
C PHI = PF. WHEN NBDCND = 2 OR 3,  
C  
C BDPF(I) = (D/DPHI)U(THETA(I),PF),  
C I = 1,2,...,M+1 .  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDPF IS  
C A DUMMY VARIABLE.  
C  
C ELMBDA  
C THE CONSTANT LAMBDA IN THE HELMHOLTZ  
C EQUATION. IF LAMBDA .GT. 0, A SOLUTION  
C MAY NOT EXIST. HOWEVER, HWSSSP WILL  
C ATTEMPT TO FIND A SOLUTION.  
C  
C F  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUE OF THE RIGHT SIDE OF THE HELMHOLTZ  
C EQUATION AND BOUNDARY VALUES (IF ANY).  
C F MUST BE DIMENSIONED AT LEAST (M+1)\*(N+1).  
C  
C ON THE INTERIOR, F IS DEFINED AS FOLLOWS:  
C FOR I = 2,3,...,M AND J = 2,3,...,N  
C F(I,J) = F(THETA(I),PHI(J)).  
C  
C ON THE 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 ------ ------------ ------------  
C  
C 1 U(TS,PHI(J)) U(TF,PHI(J))  
C 2 U(TS,PHI(J)) F(TF,PHI(J))  
C 3 F(TS,PHI(J)) F(TF,PHI(J))  
C 4 F(TS,PHI(J)) U(TF,PHI(J))  
C 5 F(0,PS) U(TF,PHI(J))  
C 6 F(0,PS) F(TF,PHI(J))  
C 7 U(TS,PHI(J)) F(PI,PS)  
C 8 F(TS,PHI(J)) F(PI,PS)  
C 9 F(0,PS) F(PI,PS)  
C  
C NBDCND F(I,1) F(I,N+1)  
C ------ -------------- --------------  
C  
C 0 F(THETA(I),PS) F(THETA(I),PS)  
C 1 U(THETA(I),PS) U(THETA(I),PF)  
C 2 U(THETA(I),PS) F(THETA(I),PF)  
C 3 F(THETA(I),PS) F(THETA(I),PF)  
C 4 F(THETA(I),PS) U(THETA(I),PF)  
C  
C NOTE:  
C IF THE TABLE CALLS FOR BOTH THE SOLUTION U  
C AND THE RIGHT SIDE F AT A CORNER THEN THE  
C 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  
C HWSSSP. THIS PARAMETER IS USED TO SPECIFY  
C THE VARIABLE DIMENSION OF F. IDIMF MUST BE  
C AT LEAST M+1 .  
C  
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)), I = 1,2,...,M+1 AND  
C J = 1,2,...,N+1 .  
C  
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  
C THIS SOLUTION, WHICH IS A LEAST SQUARES  
C SOLUTION TO THE ORIGINAL APPROXIMATION.  
C THIS SOLUTION IS NOT UNIQUE AND IS  
C UNNORMALIZED. THE VALUE OF PERTRB SHOULD  
C BE SMALL COMPARED TO THE RIGHT SIDE F.  
C 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  
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  
C = 2 TS.GE.TF  
C = 3 MBDCND.LT.1 OR MBDCND.GT.9  
C = 4 PS.LT.0 OR PS.GT.PI+PI  
C = 5 PS.GE.PF  
C = 6 N.LT.5  
C = 7 M.LT.5  
C = 8 NBDCND.LT.0 OR NBDCND.GT.4  
C = 9 ELMBDA.GT.0  
C = 10 IDIMF.LT.M+1  
C = 11 NBDCND EQUALS 1,2 OR 4 AND MBDCND.GE.5  
C = 12 TS.EQ.0 AND MBDCND EQUALS 3,4 OR 8  
C = 13 TF.EQ.PI AND MBDCND EQUALS 2,3 OR 6  
C = 14 MBDCND EQUALS 5,6 OR 9 AND TS.NE.0  
C = 15 MBDCND.GE.7 AND TF.NE.PI  
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 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  
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.  
c Revised in June 2004 by John Adams using  
c Fortran 90 dynamically allocated work space.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM THE ROUTINE DEFINES THE FINITE DIFFERENCE  
C EQUATIONS, INCORPORATES BOUNDARY DATA, AND  
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)  
C BUT ALSO DEPENDS ON INPUT PARAMETERS NBDCND  
C AND MBDCND.  
C  
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.  
C  
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),  
C PP 212-215.  
C  
C SWARZTRAUBER,P. AND R. SWEET, "EFFICIENT  
C FORTRAN SUBPROGRAMS FOR THE SOLUTION OF  
C ELLIPTIC EQUATIONS", NCAR TN/IA-109, JULY,  
C 1975, 138 PP.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**POIS3D**

C  
C file pois3d.txt (documentation for the FISHPACK solver POIS3D)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE POIS3D (LPEROD,L,C1,MPEROD,M,C2,NPEROD,N,A,B,C,LDIMF,  
C + MDIMF,F,IERROR)  
C  
C  
C DIMENSION OF A(N), B(N), C(N), F(LDIMF,MDIMF,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE LINEAR SYSTEM OF EQUATIONS  
C FOR UNKNOWN X VALUES, WHERE I=1,2,...,L,  
C J=1,2,...,M, AND K=1,2,...,N  
C  
C C1\*(X(I-1,J,K) -2.\*X(I,J,K) +X(I+1,J,K)) +  
C C2\*(X(I,J-1,K) -2.\*X(I,J,K) +X(I,J+1,K)) +  
C A(K)\*X(I,J,K-1) +B(K)\*X(I,J,K)+ C(K)\*X(I,J,K+1)  
C = F(I,J,K)  
C  
C THE INDICES K-1 AND K+1 ARE EVALUATED MODULO N,  
C I.E. X(I,J,0)=X(I,J,N) AND X(I,J,N+1)=X(I,J,1).  
C THE UNKNOWNS  
C X(0,J,K), X(L+1,J,K), X(I,0,K), AND X(I,M+1,K)  
C ARE ASSUMED TO TAKE ON CERTAIN PRESCRIBED  
C VALUES DESCRIBED BELOW.  
C  
C USAGE CALL POIS3D (LPEROD,L,C1,MPEROD,M,C2,NPEROD,  
C N,A,B,C,LDIMF,MDIMF,F,IERROR)  
C  
C ARGUMENTS  
C  
C ON INPUT  
C LPEROD  
C INDICATES THE VALUES THAT X(0,J,K) AND  
C X(L+1,J,K) ARE ASSUMED TO HAVE.  
C = 0 X(0,J,K)=X(L,J,K), X(L+1,J,K)=X(1,J,K)  
C = 1 X(0,J,K) = 0, X(L+1,J,K) = 0  
C = 2 X(0,J,K)=0, X(L+1,J,K)=X(L-1,J,K)  
C = 3 X(0,J,K)=X(2,J,K), X(L+1,J,K)=X(L-1,J,K)  
C = 4 X(0,J,K)=X(2,J,K), X(L+1,J,K) = 0.  
C  
C L  
C THE NUMBER OF UNKNOWNS IN THE I-DIRECTION.  
C L MUST BE AT LEAST 3.  
C  
C C1  
C REAL CONSTANT IN THE ABOVE LINEAR SYSTEM  
C OF EQUATIONS TO BE SOLVED.  
C  
C MPEROD  
C INDICATES THE VALUES THAT X(I,0,K) AND  
C X(I,M+1,K) ARE ASSUMED TO HAVE.  
C = 0 X(I,0,K)=X(I,M,K), X(I,M+1,K)=X(I,1,K)  
C = 1 X(I,0,K)=0, X(I,M+1,K)=0  
C = 2 X(I,0,K)=0, X(I,M+1,K)=X(I,M-1,K)  
C = 3 X(I,0,K)=X(I,2,K) X(I,M+1,K)=X(I,M-1,K)  
C = 4 X(I,0,K)=X(I,2,K) X(I,M+1,K)=0  
C  
C M  
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.  
C M MUST BE AT LEAST 3.  
C  
C C2  
C REAL CONSTANT IN THE ABOVE LINEAR SYSTEM  
C OF EQUATIONS TO BE SOLVED.  
C  
C NPEROD  
C = 0 IF A(1) AND C(N) ARE NOT ZERO.  
C = 1 IF A(1) = C(N) = 0.  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE K-DIRECTION.  
C N MUST BE AT LEAST 3.  
C  
C A, B, C  
C ONE-DIMENSIONAL ARRAYS OF LENGTH N THAT  
C SPECIFY THE COEFFICIENTS IN THE LINEAR  
C EQUATIONS GIVEN ABOVE.  
C  
C IF NPEROD = 0 THE ARRAY ELEMENTS MUST NOT  
C DEPEND UPON INDEX K, BUT MUST BE CONSTANT.  
C SPECIFICALLY,THE SUBROUTINE CHECKS THE  
C FOLLOWING CONDITION  
C A(K) = C(1)  
C C(K) = C(1)  
C B(K) = B(1)  
C FOR K=1,2,...,N.  
C  
C LDIMF  
C THE ROW (OR FIRST) DIMENSION OF THE THREE-  
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. LDIMF MUST BE AT LEAST L.  
C  
C MDIMF  
C THE COLUMN (OR SECOND) DIMENSION OF THE THREE  
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  
C A THREE-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE RIGHT SIDE OF THE LINEAR SYSTEM  
C OF EQUATIONS GIVEN ABOVE. F MUST BE  
C DIMENSIONED AT LEAST L X M X N.  
C  
C ON OUTPUT  
C  
C F  
C CONTAINS THE SOLUTION X.  
C  
C IERROR  
C AN ERROR FLAG THAT INDICATES INVALID INPUT  
C PARAMETERS. 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. 4  
C = 4 IF M .LT. 3  
C = 5 IF NPEROD .LT. 0 OR .GT. 1  
C = 6 IF N .LT. 3  
C = 7 IF LDIMF .LT. L  
C = 8 IF MDIMF .LT. M  
C = 9 IF A(K) .NE. C(1) OR C(K) .NE. C(1)  
C OR B(I) .NE.B(1) FOR SOME K=1,2,...,N.  
C = 10 IF NPEROD = 1 AND A(1) .NE. 0  
C OR C(N) .NE. 0  
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 SINCE THIS IS THE ONLY MEANS OF INDICATING A  
C POSSIBLY INCORRECT CALL TO POIS3D, THE USER  
C SHOULD TEST IERROR AFTER THE CALL.  
C  
C SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED files fish.f,comf.f,fftpack.f  
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.  
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 SOLVES THREE-DIMENSIONAL BLOCK  
C TRIDIAGONAL LINEAR SYSTEMS ARISING FROM FINITE  
C DIFFERENCE APPROXIMATIONS TO THREE-DIMENSIONAL  
C 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.  
C  
C ACCURACY TO MEASURE THE ACCURACY OF THE ALGORITHM A  
C UNIFORM RANDOM NUMBER GENERATOR WAS USED TO  
C 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  
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 APPROXIMATE SOLUTION Z. RELATIVE ERROR  
C  
C E = MAX(ABS(Z(I,J,K)-X(I,J,K)))/MAX(ABS(X(I,J,K  
C  
C WAS COMPUTED, WHERE THE TWO MAXIMA ARE TAKEN  
C OVER I=1,2,...,L, J=1,2,...,M AND K=1,2,...,N.  
C VALUES OF E ARE GIVEN IN THE TABLE BELOW FOR  
C SOME TYPICAL VALUES OF L,M AND N.  
C  
C L(=M=N) LPEROD MPEROD E  
C ------ ------ ------ ------  
C  
C 16 0 0 1.E-13  
C 15 1 1 4.E-13  
C 17 3 3 2.E-13  
C 32 0 0 2.E-13  
C 31 1 1 2.E-12  
C 33 3 3 7.E-13  
C  
C REFERENCES NONE  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**POISTG**

C  
C file poistg.txt (documentation for the FISHPACK solver POISTG)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE POISTG (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,IERROR)  
C  
C  
C DIMENSION OF A(M), B(M), C(M), Y(IDIMY,N)  
C ARGUMENTS  
C  
C LATEST REVISION June 2004  
C  
C PURPOSE SOLVES THE LINEAR SYSTEM OF EQUATIONS  
C FOR UNKNOWN X VALUES, WHERE I=1,2,...,M  
C AND J=1,2,...,N  
C  
C A(I)\*X(I-1,J) + B(I)\*X(I,J) + C(I)\*X(I+1,J)  
C + X(I,J-1) - 2.\*X(I,J) + X(I,J+1)  
C = Y(I,J)  
C  
C THE INDICES I+1 AND I-1 ARE EVALUATED MODULO M,  
C I.E. X(0,J) = X(M,J) AND X(M+1,J) = X(1,J), AND  
C X(I,0) MAY BE EQUAL TO X(I,1) OR -X(I,1), AND  
C X(I,N+1) MAY BE EQUAL TO X(I,N) OR -X(I,N),  
C DEPENDING ON AN INPUT PARAMETER.  
C  
C USAGE CALL POISTG (NPEROD,N,MPEROD,M,A,B,C,IDIMY,Y,  
C IERROR)  
C  
C ARGUMENTS  
C  
C ON INPUT  
C  
C NPEROD  
C INDICATES VALUES WHICH X(I,0) AND X(I,N+1)  
C ARE ASSUMED TO HAVE.  
C = 1 IF X(I,0) = -X(I,1) AND X(I,N+1) = -X(I,N  
C = 2 IF X(I,0) = -X(I,1) AND X(I,N+1) = X(I,N  
C = 3 IF X(I,0) = X(I,1) AND X(I,N+1) = X(I,N  
C = 4 IF X(I,0) = X(I,1) AND X(I,N+1) = -X(I,N  
C  
C N  
C THE NUMBER OF UNKNOWNS IN THE J-DIRECTION.  
C N MUST BE GREATER THAN 2.  
C  
C MPEROD  
C = 0 IF A(1) AND C(M) ARE NOT ZERO  
C = 1 IF A(1) = C(M) = 0  
C  
C M  
C THE NUMBER OF UNKNOWNS IN THE I-DIRECTION.  
C M MUST BE GREATER THAN 2.  
C  
C A,B,C  
C ONE-DIMENSIONAL ARRAYS OF LENGTH M THAT  
C SPECIFY THE COEFFICIENTS IN THE LINEAR  
C EQUATIONS GIVEN ABOVE. IF MPEROD = 0 THE  
C ARRAY ELEMENTS MUST NOT DEPEND ON INDEX I,  
C BUT MUST BE CONSTANT. SPECIFICALLY, THE  
C SUBROUTINE CHECKS THE FOLLOWING CONDITION  
C A(I) = C(1)  
C B(I) = B(1)  
C C(I) = C(1)  
C FOR I = 1, 2, ..., M.  
C  
C IDIMY  
C THE ROW (OR FIRST) DIMENSION OF THE TWO-  
C DIMENSIONAL ARRAY Y AS IT APPEARS IN THE  
C PROGRAM CALLING POISTG. THIS PARAMETER IS  
C USED TO SPECIFY THE VARIABLE DIMENSION OF Y.  
C IDIMY MUST BE AT LEAST M.  
C  
C Y  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE RIGHT SIDE OF THE LINEAR SYSTEM  
C OF EQUATIONS GIVEN ABOVE.  
C Y MUST BE DIMENSIONED AT LEAST M X N.  
C  
C ON OUTPUT  
C  
C Y  
C CONTAINS THE SOLUTION X.  
C  
C IERROR  
C AN ERROR FLAG THAT INDICATES INVALID INPUT  
C PARAMETERS. EXCEPT FOR NUMBER ZERO, A  
C SOLUTION IS NOT ATTEMPTED.  
C = 0 NO ERROR  
C = 1 IF M .LE. 2  
C = 2 IF N .LE. 2  
C = 3 IDIMY .LT. M  
C = 4 IF NPEROD .LT. 1 OR NPEROD .GT. 4  
C = 5 IF MPEROD .LT. 0 OR MPEROD .GT. 1  
C = 6 IF MPEROD = 0 AND A(I) .NE. C(1)  
C OR B(I) .NE. B(1) OR C(I) .NE. C(1)  
C FOR SOME I = 1, 2, ..., M.  
C = 7 IF MPEROD .EQ. 1 .AND.  
C (A(1).NE.0 .OR. C(M).NE.0)  
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 SINCE THIS IS THE ONLY MEANS OF INDICATING A  
C POSSIBLY INCORRECT CALL TO POIS3D, THE USER  
C SHOULD TEST IERROR AFTER THE CALL.  
C  
C  
C  
C I/O NONE  
C  
C PRECISION SINGLE  
C  
C REQUIRED files fish.f,gnbnaux.f,comf.f  
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.  
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 IS AN IMPLEMENTATION OF THE  
C ALGORITHM PRESENTED IN THE REFERENCE BELOW.  
C  
C TIMING FOR LARGE M AND N, THE EXECUTION TIME IS  
C ROUGHLY PROPORTIONAL TO M\*N\*LOG2(N).  
C  
C ACCURACY TO MEASURE THE ACCURACY OF THE ALGORITHM A  
C UNIFORM RANDOM NUMBER GENERATOR WAS USED TO  
C CREATE A SOLUTION ARRAY X FOR THE SYSTEM GIVEN  
C IN THE 'PURPOSE' SECTION ABOVE, WITH  
C A(I) = C(I) = -0.5\*B(I) = 1, I=1,2,...,M  
C AND, WHEN MPEROD = 1  
C A(1) = C(M) = 0  
C B(1) = B(M) =-1.  
C  
C THE SOLUTION X WAS SUBSTITUTED INTO THE GIVEN  
C SYSTEM AND, USING DOUBLE PRECISION, A RIGHT SID  
C Y WAS COMPUTED. USING THIS ARRAY Y SUBROUTINE  
C POISTG WAS CALLED TO PRODUCE AN APPROXIMATE  
C SOLUTION Z. THEN THE RELATIVE ERROR, DEFINED A  
C E = MAX(ABS(Z(I,J)-X(I,J)))/MAX(ABS(X(I,J)))  
C WHERE THE TWO MAXIMA ARE TAKEN OVER I=1,2,...,M  
C AND J=1,2,...,N, WAS COMPUTED. VALUES OF E ARE  
C GIVEN IN THE TABLE BELOW FOR SOME TYPICAL VALUE  
C OF M AND N.  
C  
C M (=N) MPEROD NPEROD E  
C ------ ------ ------ ------  
C  
C 31 0-1 1-4 9.E-13  
C 31 1 1 4.E-13  
C 31 1 3 3.E-13  
C 32 0-1 1-4 3.E-12  
C 32 1 1 3.E-13  
C 32 1 3 1.E-13  
C 33 0-1 1-4 1.E-12  
C 33 1 1 4.E-13  
C 33 1 3 1.E-13  
C 63 0-1 1-4 3.E-12  
C 63 1 1 1.E-12  
C 63 1 3 2.E-13  
C 64 0-1 1-4 4.E-12  
C 64 1 1 1.E-12  
C 64 1 3 6.E-13  
C 65 0-1 1-4 2.E-13  
C 65 1 1 1.E-11  
C 65 1 3 4.E-13  
C  
C REFERENCES SCHUMANN, U. AND R. SWEET,"A DIRECT METHOD  
C FOR THE SOLUTION OF POISSON"S EQUATION WITH  
C NEUMANN BOUNDARY CONDITIONS ON A STAGGERED  
C GRID OF ARBITRARY SIZE," J. COMP. PHYS.  
C 20(1976), PP. 171-182.  
C \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**SEPELI**

C  
C file sepeli.txt (documentation for the FISHPACK solver SEPELI)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE SEPELI (INTL,IORDER,A,B,M,MBDCND,BDA,ALPHA,BDB,BETA,C,  
C + D,N,NBDCND,BDC,GAMA,BDD,XNU,COFX,COFY,GRHS,  
C + USOL,IDMN,W,PERTRB,IERROR)  
C  
C  
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 LATEST REVISION JUNE 2004  
C  
C PURPOSE SEPELI SOLVES FOR EITHER THE SECOND-ORDER  
C FINITE DIFFERENCE APPROXIMATION OR A  
C FOURTH-ORDER APPROXIMATION TO A SEPARABLE  
C ELLIPTIC EQUATION  
C  
C 2 2  
C AF(X)\*D U/DX + BF(X)\*DU/DX + CF(X)\*U +  
C 2 2  
C DF(Y)\*D U/DY + EF(Y)\*DU/DY + FF(Y)\*U  
C  
C = G(X,Y)  
C  
C ON A RECTANGLE (X GREATER THAN OR EQUAL TO A  
C AND LESS THAN OR EQUAL TO B; Y GREATER THAN  
C OR EQUAL TO C AND LESS THAN OR EQUAL TO D).  
C ANY COMBINATION OF PERIODIC OR MIXED BOUNDARY  
C CONDITIONS IS ALLOWED.  
C  
C THE POSSIBLE BOUNDARY CONDITIONS ARE:  
C IN THE X-DIRECTION:  
C (0) PERIODIC, U(X+B-A,Y)=U(X,Y) FOR ALL  
C Y,X (1) U(A,Y), U(B,Y) ARE SPECIFIED FOR  
C ALL Y  
C (2) U(A,Y), DU(B,Y)/DX+BETA\*U(B,Y) ARE  
C SPECIFIED FOR ALL Y  
C (3) DU(A,Y)/DX+ALPHA\*U(A,Y),DU(B,Y)/DX+  
C BETA\*U(B,Y) ARE SPECIFIED FOR ALL Y  
C (4) DU(A,Y)/DX+ALPHA\*U(A,Y),U(B,Y) ARE  
C SPECIFIED FOR ALL Y  
C  
C IN THE Y-DIRECTION:  
C (0) PERIODIC, U(X,Y+D-C)=U(X,Y) FOR ALL X,Y  
C (1) U(X,C),U(X,D) ARE SPECIFIED FOR ALL X  
C (2) U(X,C),DU(X,D)/DY+XNU\*U(X,D) ARE  
C SPECIFIED FOR ALL X  
C (3) DU(X,C)/DY+GAMA\*U(X,C),DU(X,D)/DY+  
C XNU\*U(X,D) ARE SPECIFIED FOR ALL X  
C (4) DU(X,C)/DY+GAMA\*U(X,C),U(X,D) ARE  
C SPECIFIED FOR ALL X  
C  
C USAGE CALL SEPELI (INTL,IORDER,A,B,M,MBDCND,BDA,  
C ALPHA,BDB,BETA,C,D,N,NBDCND,BDC,  
C GAMA,BDD,XNU,COFX,COFY,GRHS,USOL,  
C IDMN,W,PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT INTL  
C = 0 ON INITIAL ENTRY TO SEPELI OR IF ANY  
C OF THE ARGUMENTS C,D, N, NBDCND, COFY  
C ARE CHANGED FROM A PREVIOUS CALL  
C = 1 IF C, D, N, NBDCND, COFY ARE UNCHANGED  
C FROM THE PREVIOUS CALL.  
C  
C IORDER  
C = 2 IF A SECOND-ORDER APPROXIMATION  
C IS SOUGHT  
C = 4 IF A FOURTH-ORDER APPROXIMATION  
C IS SOUGHT  
C  
C A,B  
C THE RANGE OF THE X-INDEPENDENT VARIABLE,  
C I.E., X IS GREATER THAN OR EQUAL TO A  
C AND LESS THAN OR EQUAL TO B. A MUST BE  
C LESS THAN B.  
C  
C M  
C 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-  
C DIRECTION GIVEN BY XI=A+(I-1)\*DLX  
C FOR I=1,2,...,M+1 WHERE DLX=(B-A)/M IS  
C THE PANEL WIDTH. M MUST BE LESS THAN  
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  
C SPECIFIED FOR ALL Y  
C = 2 IF THE SOLUTION IS SPECIFIED AT X=A AND  
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  
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  
C THAT 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. WHEN MBDCND HAS ANY OTHER  
C OTHER VALUE, BDA IS A DUMMY PARAMETER.  
C  
C ALPHA  
C THE SCALAR MULTIPLYING THE SOLUTION IN  
C CASE OF A MIXED BOUNDARY CONDITION AT X=A  
C (SEE ARGUMENT BDA). IF MBDCND IS NOT  
C EQUAL TO 3 OR 4 THEN ALPHA IS A DUMMY  
C PARAMETER.  
C  
C BDB  
C A ONE-DIMENSIONAL ARRAY OF LENGTH N+1  
C THAT SPECIFIES THE VALUES OF  
C DU(B,Y)/DX+ BETA\*U(B,Y) AT X=B.  
C WHEN MBDCND=2 OR 3  
C BDB(J) = DU(B,YJ)/DX+BETA\*U(B,YJ),  
C J=1,2,...,N+1. WHEN MBDCND HAS ANY OTHER  
C 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  
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  
C AND LESS THAN OR EQUAL TO D. C MUST BE  
C LESS THAN D.  
C  
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 DLY=(D-C)/N IS THE PANEL WIDTH.  
C IN ADDITION, N MUST BE GREATER THAN 4.  
C  
C NBDCND  
C INDICATES THE TYPES OF BOUNDARY CONDITIONS  
C AT Y=C AND Y=D  
C  
C = 0 IF THE SOLUTION IS 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 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., U(X,C) AND  
C DU(X,D)/DY+XNU\*U(X,D) ARE SPECIFIED  
C FOR ALL X  
C = 3 IF THE BOUNDARY CONDITIONS ARE MIXED  
C AT Y=C AND Y=D, I.E.,  
C 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 = 4 IF THE BOUNDARY CONDITION IS MIXED  
C AT Y=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  
C BDC  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1  
C THAT SPECIFIES THE VALUE OF  
C DU(X,C)/DY+GAMA\*U(X,C) AT Y=C.  
C WHEN NBDCND=3 OR 4 BDC(I) = DU(XI,C)/DY +  
C GAMA\*U(XI,C), I=1,2,...,M+1.  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC  
C IS A DUMMY PARAMETER.  
C  
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  
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.  
C 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 VALUE, BDD  
C IS A DUMMY PARAMETER.  
C  
C 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  
C DUMMY PARAMETER.  
C  
C COFX  
C A USER-SUPPLIED SUBPROGRAM WITH  
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  
C A USER-SUPPLIED SUBPROGRAM WITH PARAMETERS  
C Y, DFUN, EFUN, FFUN WHICH RETURNS THE  
C VALUES OF THE Y-DEPENDENT COEFFICIENTS  
C 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  
C  
C MBDCND GRHS(1,J) GRHS(M+1,J)  
C ------ --------- -----------  
C 0 G(A,YJ) G(B,YJ)  
C 1 \* \*  
C 2 \* G(B,YJ) J=1,2,...,N+1  
C 3 G(A,YJ) G(B,YJ)  
C 4 G(A,YJ) \*  
C  
C NBDCND GRHS(I,1) GRHS(I,N+1)  
C ------ --------- -----------  
C 0 G(XI,C) G(XI,D)  
C 1 \* \*  
C 2 \* G(XI,D) I=1,2,...,M+1  
C 3 G(XI,C) G(XI,D)  
C 4 G(XI,C) \*  
C  
C WHERE \* MEANS THESE QUANTITIES ARE NOT USED.  
C GRHS SHOULD BE DIMENSIONED IDMN BY AT LEAST  
C N+1 IN THE CALLING ROUTINE.  
C  
C USOL  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE SOLUTION ALONG THE BOUNDARIES.  
C AT THE BOUNDARIES, USOL IS 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,YJ)  
C  
C NBDCND USOL(I,1) USOL(I,N+1)  
C ------ --------- -----------  
C 0 \* \*  
C 1 U(XI,C) U(XI,D)  
C 2 U(XI,C) \* I=1,2,...,M+1  
C 3 \* \*  
C 4 \* U(XI,D)  
C  
C WHERE \* MEANS THE QUANTITIES ARE NOT USED  
C IN THE SOLUTION.  
C  
C IF IORDER=2, THE USER MAY EQUIVALENCE GRHS  
C AND USOL TO SAVE SPACE. NOTE THAT IN THIS  
C CASE THE TABLES SPECIFYING THE BOUNDARIES  
C OF THE GRHS AND USOL ARRAYS DETERMINE THE  
C BOUNDARIES 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  
C BE CHOSEN. FOR EXAMPLE, IF MBDCND=2 AND  
C NBDCND=4, THEN U(A,C), U(A,D), U(B,D) MUST  
C BE CHOSEN AT THE CORNERS IN ADDITION  
C 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 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 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  
c USE fish  
c  
c The declarative statement  
c  
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 variable (defined in  
c the module "fish.f") which is used internally  
c in SEPELI 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  
c 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 SEPELI. These  
c must be preserved on non-initial calls (INTL=1)  
c to SEPELI. This eliminates redundant calculations  
c and saves compute time.  
c \*\*\*\* IMPORTANT! The user program calling SEPELI should  
c include the statement:  
c  
c CALL FISHFIN(W)  
C  
C after the final approximation is generated by  
C SEPELI. The will deallocate the real and complex  
c work space of W. Failure to include this statement  
c could result in serious memory leakage.  
c  
C  
C ON OUTPUT USOL  
C CONTAINS THE APPROXIMATE SOLUTION TO THE  
C ELLIPTIC EQUATION.  
C USOL(I,J) IS THE APPROXIMATION TO U(XI,YJ)  
C FOR I=1,2...,M+1 AND J=1,2,...,N+1.  
C THE APPROXIMATION HAS ERROR  
C O(DLX\*\*2+DLY\*\*2) IF CALLED WITH IORDER=2  
C AND O(DLX\*\*4+DLY\*\*4) IF CALLED WITH  
C IORDER=4.  
C  
C W  
c The derived type (fishworkspace) variable W  
c contains real and complex values that must not  
C be destroyed if SEPELI is called again with  
C INTL=1.  
C  
C PERTRB  
C IF A COMBINATION OF PERIODIC OR DERIVATIVE  
C BOUNDARY CONDITIONS  
C (I.E., ALPHA=BETA=0 IF MBDCND=3;  
C GAMA=XNU=0 IF NBDCND=3) IS SPECIFIED  
C AND IF THE COEFFICIENTS OF U(X,Y) IN THE  
C SEPARABLE ELLIPTIC EQUATION ARE ZERO  
C (I.E., CF(X)=0 FOR X GREATER THAN OR EQUAL  
C TO A AND LESS THAN OR EQUAL TO B;  
C FF(Y)=0 FOR Y GREATER THAN OR EQUAL TO C  
C AND LESS THAN OR EQUAL TO D) THEN A  
C SOLUTION MAY NOT EXIST. PERTRB IS A  
C CONSTANT CALCULATED AND SUBTRACTED FROM  
C THE RIGHT-HAND SIDE OF THE MATRIX EQUATIONS  
C GENERATED BY SEPELI WHICH INSURES THAT A  
C SOLUTION EXISTS. SEPELI 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  
C = 0 NO ERROR  
C = 1 IF A GREATER THAN B OR C GREATER THAN D  
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)  
C = 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)  
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  
C FOR SOME INTERIOR MESH POINT (XI,YJ)  
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  
C NOTE (CONCERNING IERROR=4): FOR THE  
C COEFFICIENTS INPUT THROUGH COFX, COFY,  
C THE DISCRETIZATION MAY LEAD TO A BLOCK  
C TRIDIAGONAL LINEAR SYSTEM WHICH IS NOT  
C DIAGONALLY DOMINANT (FOR EXAMPLE, THIS  
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  
C PRECISION 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  
C LIBRARIES IN JANUARY 1980. Revised in June  
C 2004 using Fortan 90 dynamically allocated work  
c space and derived data types to eliminate mixed  
c mode conflicts in the earlier versions. All  
c statement labels, arithmetic if statements and  
c computed GO TO statements have been removed from  
c the current version of SEPELI.  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM SEPELI AUTOMATICALLY DISCRETIZES THE  
C SEPARABLE ELLIPTIC EQUATION WHICH IS THEN  
C SOLVED BY A GENERALIZED CYCLIC REDUCTION  
C ALGORITHM IN THE SUBROUTINE, BLKTRI. THE  
C FOURTH-ORDER SOLUTION IS OBTAINED USING  
C 'DEFERRED CORRECTIONS' WHICH IS DESCRIBED  
C AND REFERENCED IN SECTIONS, REFERENCES AND  
C METHOD.  
C  
C TIMING THE OPERATIONAL COUNT IS PROPORTIONAL TO  
C 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  
C UNTIL THE MESH IS sufficiently refined.  
C  
C SECOND-ORDER FOURTH-ORDER  
C M N ERROR ERROR  
C  
C 6 6 6.8E-1 1.2E0  
C 14 14 1.4E-1 1.8E-1  
C 30 30 3.2E-2 9.7E-3  
C 62 62 7.5E-3 3.0E-4  
C 126 126 1.8E-3 3.5E-6  
C  
C  
C REFERENCES KELLER, H.B., NUMERICAL METHODS FOR TWO-POINT  
C BOUNDARY-VALUE PROBLEMS, BLAISDEL (1968),  
C WALTHAM, MASS.  
C  
C SWARZTRAUBER, P., AND R. SWEET (1975):  
C EFFICIENT FORTRAN SUBPROGRAMS FOR THE  
C SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL  
C EQUATIONS. NCAR TECHNICAL NOTE  
C NCAR-TN/IA-109, PP. 135-137.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**SEPX4**

C  
C file sepx4.txt (documentation for the FISHPACK solver SEPX4)  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C SUBROUTINE SEPX4 (IORDER,A,B,M,MBDCND,BDA,ALPHA,BDB,BETA,C,D,N,  
C + NBDCND,BDC,BDD,COFX,GRHS,USOL,IDMN,PERTRB,  
C + IERROR)  
C  
C  
C  
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  
C PURPOSE SEPX4 SOLVES FOR EITHER THE SECOND-ORDER  
C FINITE DIFFERENCE APPROXIMATION OR A  
C FOURTH-ORDER APPROXIMATION TO A SEPARABLE  
C ELLIPTIC EQUATION  
C  
C AF(X)\*UXX+BF(X)\*UX+CF(X)\*U+UYY = G(X,Y)  
C  
C ON A RECTANGLE (X GREATER THAN OR EQUAL TO  
C A AND LESS THAN OR EQUAL TO B, Y GREATER THAN  
C OR EQUAL TO C AND LESS THAN OR EQUAL TO D).  
C ANY COMBINATION OF PERIODIC OR MIXED BOUNDARY  
C CONDITIONS IS ALLOWED. IF BOUNDARY  
C CONDITIONS IN THE X DIRECTION ARE PERIODIC  
C (SEE MBDCND=0 BELOW) THEN THE COEFFICIENTS  
C MUST SATISFY  
C  
C AF(X)=C1,BF(X)=0,CF(X)=C2 FOR ALL X.  
C  
C HERE C1,C2 ARE CONSTANTS, C1.GT.0.  
C  
C THE POSSIBLE BOUNDARY CONDITIONS ARE:  
C IN THE X-DIRECTION:  
C (0) PERIODIC, U(X+B-A,Y)=U(X,Y) FOR  
C ALL Y,X  
C (1) U(A,Y), U(B,Y) ARE SPECIFIED FOR ALL Y  
C (2) U(A,Y), DU(B,Y)/DX+BETA\*U(B,Y) ARE  
C SPECIFIED FOR ALL Y  
C (3) DU(A,Y)/DX+ALPHA\*U(A,Y),DU(B,Y)/DX+  
C BETA\*U(B,Y) ARE SPECIFIED FOR ALL Y  
C (4) DU(A,Y)/DX+ALPHA\*U(A,Y),U(B,Y) ARE  
C SPECIFIED FOR ALL Y  
C  
C IN THE Y-DIRECTION:  
C (0) PERIODIC, U(X,Y+D-C)=U(X,Y) FOR ALL X,Y  
C (1) U(X,C),U(X,D) ARE SPECIFIED FOR ALL X  
C (2) U(X,C),DU(X,D)/DY ARE SPECIFIED FOR  
C ALL X  
C (3) DU(X,C)/DY,DU(X,D)/DY ARE SPECIFIED FOR  
C ALL X  
C (4) DU(X,C)/DY,U(X,D) ARE SPECIFIED FOR  
C ALL X  
C  
C USAGE CALL SEPX4(IORDER,A,B,M,MBDCND,BDA,ALPHA,BDB,  
C BETA,C,D,N,NBDCND,BDC,BDD,COFX,  
C GRHS,USOL,IDMN,W,PERTRB,IERROR)  
C  
C ARGUMENTS  
C ON INPUT IORDER  
C = 2 IF A SECOND-ORDER APPROXIMATION IS  
C SOUGHT  
C = 4 IF A FOURTH-ORDER APPROXIMATION IS  
C SOUGHT  
C  
c \*\*\* caution \*\*\* GRHS SHOULD BE RESET IF SEPX4 WAS FIRST CALLED  
C WITH IORDER=2 AND WILL BE CALLED AGAIN WITH  
C IORDER=4. VALUES IN GRHS ARE DESTROYED BY THE  
C IORDER=2 CALL.  
C  
C  
C A,B  
C THE RANGE OF THE X-INDEPENDENT VARIABLE,  
C I.E., X IS GREATER THAN OR EQUAL TO A  
C AND LESS THAN OR EQUAL TO B. A MUST BE  
C LESS THAN B.  
C  
C M  
C 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-  
C DIRECTION GIVEN BY XI=A+(I-1)\*DLX  
C FOR I=1,2,...,M+1 WHERE DLX=(B-A)/M IS  
C THE PANEL WIDTH. M MUST BE LESS THAN  
C IDMN AND GREATER THAN 5.  
C  
C MBDCND  
C INDICATES THE TYPE OF BOUNDARY CONDITION  
C AT X=A AND X=B  
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  
C SPECIFIED FOR ALL Y  
C = 2 IF THE SOLUTION IS SPECIFIED AT X=A  
C AND THE BOUNDARY CONDITION IS MIXED AT  
C X=B, I.E., U(A,Y) AND  
C DU(B,Y)/DX+BETA\*U(B,Y) ARE SPECIFIED  
C FOR ALL Y  
C = 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  
C ALPHA  
C THE SCALAR MULTIPLYING THE SOLUTION IN CASE  
C 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  
C 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.  
C WHEN MBDCND=2 OR 3  
C BDB(J) = DU(B,YJ)/DX+BETA\*U(B,YJ),  
C J=1,2,...,N+1  
C WHEN MBDCND HAS ANY OTHER VALUE, BDB IS  
C A DUMMY PARAMETER.  
C  
C BETA  
C THE SCALAR MULTIPLYING THE SOLUTION IN  
C CASE OF A MIXED BOUNDARY CONDITION AT X=B  
C (SEE ARGUMENT BDB). IF MBDCND IS NOT EQUAL  
C TO 2 OR 3, THEN BETA IS A DUMMY PARAMETER.  
C  
C C,D  
C THE RANGE OF THE Y-INDEPENDENT VARIABLE,  
C I.E., Y IS GREATER THAN OR EQUAL TO C AND  
C LESS THAN OR EQUAL TO D. C MUST BE LESS  
C THAN D.  
C  
C N  
C THE NUMBER OF PANELS INTO WHICH THE  
C INTERVAL (C,D) IS SUBDIVIDED. HENCE,  
C THERE WILL BE N+1 GRID POINTS IN THE Y-  
C DIRECTION GIVEN BY YJ=C+(J-1)\*DLY FOR  
C J=1,2,...,N+1 WHERE DLY=(D-C)/N IS THE  
C PANEL WIDTH. IN ADDITION, N MUST BE  
C GREATER THAN 4.  
C  
C NBDCND  
C INDICATES THE TYPES OF BOUNDARY CONDITIONS  
C AT Y=C AND Y=D  
C = 0 IF THE SOLUTION IS 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 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 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  
C BDC  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIES THE VALUE DU(X,C)/DY AT Y=C.  
C  
C WHEN NBDCND=3 OR 4  
C BDC(I) = DU(XI,C)/DY I=1,2,...,M+1.  
C  
C WHEN NBDCND HAS ANY OTHER VALUE, BDC IS  
C A DUMMY PARAMETER.  
C  
C BDD  
C A ONE-DIMENSIONAL ARRAY OF LENGTH M+1 THAT  
C SPECIFIED THE VALUE OF DU(X,D)/DY AT Y=D.  
C  
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  
C COFX  
C A USER-SUPPLIED SUBPROGRAM WITH PARAMETERS  
C X, AFUN, BFUN, CFUN WHICH RETURNS THE  
C VALUES OF THE X-DEPENDENT COEFFICIENTS  
C AF(X), BF(X), CF(X) IN THE ELLIPTIC  
C EQUATION AT X. IF BOUNDARY CONDITIONS IN  
C THE X DIRECTION ARE PERIODIC THEN THE  
C COEFFICIENTS MUST SATISFY AF(X)=C1,BF(X)=0,  
C CF(X)=C2 FOR ALL X. HERE C1.GT.0  
C AND C2 ARE CONSTANTS.  
C  
C NOTE THAT COFX MUST BE DECLARED EXTERNAL  
C IN THE CALLING ROUTINE.  
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.,GRHS(I,J)=G(XI,YI),  
C FOR I=2,...,M, J=2,...,N. AT THE  
C BOUNDARIES, GRHS IS DEFINED BY  
C  
C MBDCND GRHS(1,J) GRHS(M+1,J)  
C ------ --------- -----------  
C 0 G(A,YJ) G(B,YJ)  
C 1 \* \*  
C 2 \* G(B,YJ) J=1,2,...,N+1  
C 3 G(A,YJ) G(B,YJ)  
C 4 G(A,YJ) \*  
C  
C NBDCND GRHS(I,1) GRHS(I,N+1)  
C ------ --------- -----------  
C 0 G(XI,C) G(XI,D)  
C 1 \* \*  
C 2 \* G(XI,D) I=1,2,...,M+1  
C 3 G(XI,C) G(XI,D)  
C 4 G(XI,C) \*  
C  
C WHERE \* MEANS THESE QUANTITES ARE NOT USED.  
C GRHS SHOULD BE DIMENSIONED IDMN BY AT LEAST  
C N+1 IN THE CALLING ROUTINE.  
C  
c \*\*\* caution GRHS SHOULD BE RESET IF SEPX4 WAS FIRST CALLED  
C WITH IORDER=2 AND WILL BE CALLED AGAIN WITH  
C IORDER=4. VALUES IN GRHS ARE DESTROYED BY THE  
C IORDER=2 CALL.  
C  
C USOL  
C A TWO-DIMENSIONAL ARRAY THAT SPECIFIES THE  
C VALUES OF THE SOLUTION ALONG THE BOUNDARIES.  
C AT THE BOUNDARIES, USOL IS 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,YJ)  
C  
C NBDCND USOL(I,1) USOL(I,N+1)  
C ------ --------- -----------  
C 0 \* \*  
C 1 U(XI,C) U(XI,D)  
C 2 U(XI,C) \* I=1,2,...,M+1  
C 3 \* \*  
C 4 \* U(XI,D)  
C  
C WHERE \* MEANS THE QUANTITES ARE NOT USED  
C IN THE SOLUTION.  
C  
C IF IORDER=2, THE USER MAY EQUIVALENCE GRHS  
C AND USOL TO SAVE SPACE. NOTE THAT IN THIS  
C CASE THE TABLES SPECIFYING THE BOUNDARIES  
C OF THE GRHS AND USOL ARRAYS DETERMINE THE  
C BOUNDARIES 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  
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 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  
C ON OUTPUT USOL  
C CONTAINS THE APPROXIMATE SOLUTION TO THE  
C ELLIPTIC EQUATION. USOL(I,J) IS THE  
C APPROXIMATION TO U(XI,YJ) FOR I=1,2...,M+1  
C AND J=1,2,...,N+1. THE 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  
C TO THE PDE).  
C PERTRB IS A CONSTANT CALCULATED AND  
C SUBTRACTED FROM THE RIGHT HAND SIDE OF THE  
C MATRIX EQUATION INSURING THE EXISTENCE OF A  
C SOLUTION. SEPX4 COMPUTES THIS SOLUTION  
C WHICH IS A WEIGHTED MINIMAL LEAST SQUARES  
C SOLUTION TO THE ORIGINAL PROBLEM. IF  
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  
C  
C = 0 NO ERROR  
C = 1 IF A GREATER THAN B OR C GREATER  
C 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  
C DIAGONALLY DOMINANT.)  
C = 5 IF IDMN IS TOO SMALL (SEE DISCUSSION  
C OF IDMN)  
C = 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)  
C = 8 IF IORDER IS NOT 2 OR 4  
C = 9 IF INTL IS NOT 0 OR 1  
C = 10 IF AFUN IS LESS THAN OR EQUAL TO ZERO  
C FOR SOME INTERIOR MESH POINT XI SOME  
C INTERIOR MESH POINT (XI,YJ)  
C = 12 IF MBDCND=0 AND AF(X)=CF(X)=CONSTANT  
C OR BF(X)=0 FOR ALL X IS NOT TRUE.  
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 SPECIAL CONDITIONS NONE  
C  
C I/O NONE  
C  
C REQUIRED files fish.f,comf.f,genbun.f,gnbnaux.f,sepaux.f  
C  
C  
C PRECISION SINGLE  
C  
C  
C LANGUAGE FORTRAN 90  
C  
C HISTORY SEPX4 WAS DEVELOPED AT NCAR BY JOHN C.  
C ADAMS OF THE SCIENTIFIC COMPUTING DIVISION  
C IN OCTOBER 1978. THE BASIS OF THIS CODE IS  
C NCAR ROUTINE SEPELI. BOTH PACKAGES WERE  
C RELEASED ON NCAR'S PUBLIC LIBRARIES IN  
C JANUARY 1980. SEPX4 was modified in June 2004  
c incorporating fortran 90 dynamical storage  
c allocation for work space requirements  
C  
C PORTABILITY FORTRAN 90  
C  
C ALGORITHM SEPX4 AUTOMATICALLY DISCRETIZES THE SEPARABLE  
C ELLIPTIC EQUATION WHICH IS THEN SOLVED BY A  
C GENERALIZED CYCLIC REDUCTION ALGORITHM IN THE  
C SUBROUTINE POIS. THE FOURTH ORDER SOLUTION  
C IS OBTAINED USING THE TECHNIQUE OF DEFFERRED  
C CORRECTIONS REFERENCED BELOW.  
C  
C TIMING WHEN POSSIBLE, SEPX4 SHOULD BE USED INSTEAD  
C OF PACKAGE SEPELI. THE INCREASE IN SPEED  
C IS AT LEAST A FACTOR OF THREE.  
C  
C REFERENCES KELLER, H.B., NUMERICAL METHODS FOR TWO-POINT  
C BOUNDARY-VALUE PROBLEMS, BLAISDEL (1968),  
C WALTHAM, MASS.  
C  
C SWARZTRAUBER, P., AND R. SWEET (1975):  
C EFFICIENT FORTRAN SUBPROGRAMS FOR THE  
C SOLUTION OF ELLIPTIC PARTIAL DIFFERENTIAL  
C EQUATIONS. NCAR TECHNICAL NOTE  
C NCAR-TN/IA-109, PP. 135-137.  
C\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**TBLKTRI**

C  
C file tblktri.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE BLKTRI TO  
C SOLVE THE EQUATION  
C  
C .5/S\*(D/DS)(.5/S\*DU/DS)+.5/T\*(D/DT)(.5/T\*DU/DT)  
C (1)  
C = 15/4\*S\*T\*(S\*\*4+T\*\*4)  
C  
C ON THE RECTANGLE 0 .LT. S .LT. 1 AND 0 .LT. T .LT. 1  
C WITH THE BOUNDARY CONDITIONS  
C  
C U(0,T) = 0  
C 0 .LE. T .LE. 1  
C U(1,T) = T\*\*5  
C  
C AND  
C  
C U(S,0) = 0  
C 0 .LE. S .LE. 1  
C U(S,1) = S\*\*5  
C  
C THE EXACT SOLUTION OF THIS PROBLEM IS U(S,T) = (S\*T)\*\*5  
C  
C DEFINE THE INTEGERS M = 50 AND N = 63. THEN DEFINE THE  
C GRID INCREMENTS DELTAS = 1/(M+1) AND DELTAT = 1/(N+1).  
C  
C THE GRID IS THEN GIVEN BY S(I) = I\*DELTAS FOR I = 1,...,M  
C AND T(J) = J\*DELTAT FOR J = 1,...,N.  
C  
C THE APPROXIMATE SOLUTION IS GIVEN AS THE SOLUTION TO  
C THE FOLLOWING FINITE DIFFERENCE APPROXIMATION OF EQUATION (1).  
C  
C .5/(S(I)\*DELTAS)\*((U(I+1,J)-U(I,J))/(2\*S(I+.5)\*DELTAS)  
C -(U(I,J)-U(I-1,J))/(2\*S(I-.5)\*DELTAS))  
C +.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))  
C = 15/4\*S(I)\*T(J)\*(S(I)\*\*4+T(J)\*\*4)  
C  
C WHERE S(I+.5) = .5\*(S(I+1)+S(I))  
C S(I-.5) = .5\*(S(I)+S(I-1))  
C T(I+.5) = .5\*(T(I+1)+T(I))  
C T(I-.5) = .5\*(T(I)+T(I-1))  
C  
C THE APPROACH IS TO WRITE EQUATION (2) IN THE FORM  
C  
C AM(I)\*U(I-1,J)+BM(I)\*U(I,J)+CM(I)\*U(I+1,J)  
C +AN(J)\*U(I,J-1)+BN(J)\*U(I,J)+CN(J)\*U(I,J+1) (3)  
C = Y(I,J)  
C  
C AND THEN CALL SUBROUTINE BLKTRI TO DETERMINE U(I,J)  
C  
C  
C  
 PROGRAM TBLKTRI  
 USE fish  
 implicit none  
 TYPE ( fishworkspace) :: w  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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,ERR,Z  
C-----------------------------------------------  
 IFLG = 0  
 NP = 1  
 N = 63  
 MP = 1  
 M = 50  
 IDIMY = 75  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING THE  
C 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  
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) = TEMP1\*TEMP2  
 CM(I) = TEMP1\*TEMP3  
 BM(I) = -(AM(I)+CM(I))  
 END DO  
C  
C COMPUTE THE COEFFICIENTS AN,BN,CN CORRESPONDING TO THE T DIRECTION  
C  
 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  
C COMPUTE RIGHT SIDE OF EQUATION  
C  
 DO J = 1, N  
 Y(:M,J) = 3.75\*S(:M)\*T(J)\*(S(:M)\*\*4+T(J)\*\*4)  
 END DO  
C  
C THE NONZERO BOUNDARY CONDITIONS ENTER THE LINEAR SYSTEM VIA  
C THE RIGHT SIDE Y(I,J). IF THE EQUATIONS (3) GIVEN ABOVE  
C ARE EVALUATED AT I=M AND J=1,...,N THEN THE TERM CM(M)\*U(M+1,J)  
C IS KNOWN FROM THE BOUNDARY CONDITION TO BE CM(M)\*T(J)\*\*5.  
C THEREFORE THIS TERM CAN BE INCLUDED IN THE RIGHT SIDE Y(M,J).  
C THE SAME ANALYSIS APPLIES AT J=N AND I=1,..,M. NOTE THAT THE  
C 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  
C  
C 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 <= 0)  
 CALL BLKTRI (IFLG, NP, N, AN, BN, CN, MP, M, AM, BM, CM, IDIMY  
 1 , 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**

C  
C file tcblktri.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TCBLKTRI  
 USE fish  
 implicit none  
 TYPE (fishworkspace) :: w  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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,ERR,Z  
 COMPLEX , DIMENSION(75,105) :: Y  
 COMPLEX, DIMENSION(75) :: AM, BM, CM  
C-----------------------------------------------  
C  
 IFLG = 0  
 NP = 1  
 N = 63  
 MP = 1  
 M = 50  
 IDIMY = 75  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING THE  
C 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  
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  
C  
C COMPUTE THE COEFFICIENTS AN,BN,CN CORRESPONDING TO THE T DIRECTION  
C  
 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  
C COMPUTE RIGHT SIDE OF EQUATION  
C  
 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  
C  
C THE NONZERO BOUNDARY CONDITIONS ENTER THE LINEAR SYSTEM VIA  
C THE RIGHT SIDE Y(I,J). IF THE EQUATIONS (3) GIVEN ABOVE  
C ARE EVALUATED AT I=M AND J=1,...,N THEN THE TERM CM(M)\*U(M+1,J)  
C IS KNOWN FROM THE BOUNDARY CONDITION TO BE CM(M)\*T(J)\*\*5.  
C THEREFORE THIS TERM CAN BE INCLUDED IN THE RIGHT SIDE Y(M,J).  
C THE SAME ANALYSIS APPLIES AT J=N AND I=1,..,M. NOTE THAT THE  
C 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  
 1 , 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 (\*, \*)   
 1 ' 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**

C  
C file tcmgnbn.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TCMBNGN  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 INTEGER :: IDIMF, M, MP1, MPEROD, N, NPEROD, I, J, IERROR  
 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  
C-----------------------------------------------  
c  
C PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE CMGNBN TO SOLVE  
C THE EQUATION  
C  
C (1+X)\*\*2\*(D/DX)(DU/DX) - 2(1+X)(DU/DX) + (D/DY)(DU/DY)  
C  
C - SQRT(-1)\*U = (3 - SQRT(-1))\*(1+X)\*\*4\*SIN(Y) (1)  
C  
C ON THE RECTANGLE 0 .LT. X .LT. 1 AND -PI .LT. Y .LT. PI  
C WITH THE BOUNDARY CONDITIONS  
C  
C (DU/DX)(0,Y) = 4SIN(Y) (2)  
C -PI .LE. Y .LE. PI  
C U(1,Y) = 16SIN(Y) (3)  
C  
C AND WITH U PERIODIC IN Y USING FINITE DIFFERENCES ON A  
C GRID WITH DELTAX (= DX) = 1/20 AND DELTAY (= DY) = PI/20.  
C TO SET UP THE FINITE DIFFERENCE EQUATIONS WE DEFINE  
C THE GRID POINTS  
C  
C X(I) = (I-1)DX I=1,2,...,21  
C  
C Y(J) = -PI + (J-1)DY J=1,2,...,41  
C  
C AND LET V(I,J) BE AN APPROXIMATION TO U(X(I),Y(J)).  
C NUMBERING THE GRID POINTS IN THIS FASHION GIVES THE SET  
C OF UNKNOWNS AS V(I,J) FOR I=1,2,...,20 AND J=1,2,...,40.  
C HENCE, IN THE PROGRAM M = 20 AND N = 40. AT THE INTERIOR  
C GRID POINT (X(I),Y(J)), WE REPLACE ALL DERIVATIVES IN  
C EQUATION (1) BY SECOND ORDER CENTRAL FINITE DIFFERENCES,  
C MULTIPLY BY DY\*\*2, AND COLLECT COEFFICIENTS OF V(I,J) TO  
C GET THE FINITE DIFFERENCE EQUATION  
C  
C A(I)V(I-1,J) + B(I)V(I,J) + C(I)V(I+1,J)  
C  
C + V(I,J-1) - 2V(I,J) + V(I,J+1) = F(I,J) (4)  
C  
C WHERE S = (DY/DX)\*\*2, AND FOR I=2,3,...,19  
C  
C A(I) = (1+X(I))\*\*2\*S + (1+X(I))\*S\*DX  
C  
C B(I) = -2(1+X(I))\*\*2\*S - SQRT(-1)\*DY\*\*2  
C  
C C(I) = (1+X(I))\*\*2\*S - (1+X(I))\*S\*DX  
C  
C F(I,J) = (3 - SQRT(-1))\*(1+X(I))\*\*4\*DY\*\*2\*SIN(Y(J))  
C FOR J=1,2,...,40.  
C  
C TO OBTAIN EQUATIONS FOR I = 1, WE REPLACE THE  
C DERIVATIVE IN EQUATION (2) BY A SECOND ORDER CENTRAL  
C FINITE DIFFERENCE APPROXIMATION, USE THIS EQUATION TO  
C ELIMINATE THE VIRTUAL UNKNOWN V(0,J) IN EQUATION (4)  
C AND ARRIVE AT THE EQUATION  
C  
C B(1)V(1,J) + C(1)V(2,J) + V(1,J-1) - 2V(1,J) + V(1,J+1)  
C  
C = F(1,J)  
C  
C WHERE  
C  
C B(1) = -2S - SQRT(-1)\*DY\*\*2 , C(1) = 2S  
C  
C F(1,J) = (11-SQRT(-1)+8/DX)\*DY\*\*2\*SIN(Y(J)), J=1,2,...,40.  
C  
C FOR COMPLETENESS, WE SET A(1) = 0.  
C TO OBTAIN EQUATIONS FOR I = 20, WE INCORPORATE  
C EQUATION (3) INTO EQUATION (4) BY SETTING  
C  
C V(21,J) = 16SIN(Y(J))  
C  
C AND ARRIVE AT THE EQUATION  
C  
C A(20)V(19,J) + B(20)V(20,J)  
C  
C + V(20,J-1) - 2V(20,J) + V(20,J+1) = F(20,J)  
C  
C WHERE  
C  
C A(20) = (1+X(20))\*\*2\*S + (1+X(20))\*S\*DX  
C  
C B(20) = -2\*(1+X(20))\*\*2\*S - SQRT(-1)\*DY\*\*2  
C  
C F(20,J) = ((3-SQRT(-1))\*(1+X(20))\*\*4\*DY\*\*2 - 16(1+X(20))\*\*2\*S  
C + 16(1+X(20))\*S\*DX)\*SIN(Y(J))  
C  
C FOR J=1,2,...,40.  
C  
C FOR COMPLETENESS, WE SET C(20) = 0. HENCE, IN THE  
C PROGRAM MPEROD = 1.  
C THE PERIODICITY CONDITION ON U GIVES THE CONDITIONS  
C  
C V(I,0) = V(I,40) AND V(I,41) = V(I,1) FOR I=1,2,...,20.  
C  
C HENCE, IN THE PROGRAM NPEROD = 0.  
C  
C THE EXACT SOLUTION TO THIS PROBLEM IS  
C  
C U(X,Y) = (1+X)\*\*4\*SIN(Y) .  
C  
C  
C 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  
C GENERATE GRID POINTS FOR LATER USE.  
C  
 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  
C 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  
C GENERATE RIGHT SIDE.  
C  
 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 = TSQ\*\*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  
C COMPUTE DISCRETIAZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(X,Y) = (1+X)\*\*4\*SIN(Y) .  
C  
 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**

C  
C file tgenbun.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TGENBUN  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C-----------------------------------------------  
C  
C 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  
C GENERATE GRID POINTS FOR LATER USE.  
C  
 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  
C GENERATE COEFFICIENTS.  
C  
 S = (DY/DX)\*\*2  
 DO I = 2, 19  
 T = 1. + X(I)  
 TSQ = T\*\*2  
 A(I) = (TSQ + 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.  
C  
C GENERATE RIGHT SIDE.  
C  
 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)  
 TSQ = 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  
C COMPUTE DISCRETIAZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(X,Y) = (1+X)\*\*4\*SIN(Y) .  
C  
 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**

C  
C file thstcrt.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THSTCRT  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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  
C-----------------------------------------------  
C  
C 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.  
C  
C AUXILIARY QUANTITIES  
C  
 PI = 4.0\*ATAN(1.0)  
 PISQ = PI\*PI  
C  
C GENERATE AND STORE GRID POINTS FOR COMPUTATION OF BOUNDARY DATA  
C 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  
C  
C GENERATE BOUNDARY DATA.  
C  
 DO J = 1, N  
 BDA(J) = 0.  
 BDB(J) = -PI\*COS(PI\*Y(J))  
 END DO  
C  
C BDC AND BDD ARE DUMMY ARGUMENTS IN THIS EXAMPLE.  
C  
C GENERATE RIGHT SIDE OF EQUATION.  
C  
 T = -2.\*(PISQ + 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  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(X,Y) = SIN(PI\*X)\*COS(PI\*Y) .  
C  
 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**

C  
C file thstcsp.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THSTCSP  
 USE fish  
 implicit none  
 TYPE (fishworkspace) :: w  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C NOTE THAT FROM DIMENSION STATEMENT WE GET THAT IDIMF = 47  
C  
 IDIMF = 47  
 A = 0.  
 B = 4.0\*ATAN(1.0)  
C  
C NOTE THAT B IS SET TO PI USING THE FUNCTION PIMACH AS REQUIRED.  
C  
 M = 45  
 MBDCND = 9  
 DT = (B - A)/FLOAT(M)  
C  
C DEFINE GRID POINTS THETA(I) AND COS(THETA(I))  
C  
 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)  
C  
 DO J = 1, N  
 R(J) = C + (FLOAT(J) - 0.5)\*DR  
 END DO  
C  
C DEFINE BOUNDARY ARRAY BDD. BDA, BDB, AND BDC ARE DUMMY  
C VARIABLES IN THIS EXAMPLE.  
C  
 BDD(:M) = COST(:M)\*\*4  
 ELMBDA = 0.  
C  
C 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  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(THETA,R) = (R\*COS(THETA))\*\*4  
C  
 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 ERR  
! release work space allocated by hstcsp (intl=0 call)  
 CALL FISHFIN (W)  
 STOP   
 END PROGRAM THSTCSP

**THSTCYL**

C  
C file thstcyl.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THSTCYL  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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-----------------------------------------------  
C  
C PROGRAM TO ILLUSTRATE THE USE OF HSTCYL TO SOLVE THE EQUATION  
C  
C (1/R)(D/DR)(R\*DU/DR) + (D/DZ)(DU/DZ) = (2\*R\*Z)\*\*2\*(4\*Z\*\*2 + 3\*R\*\*2)  
C  
C ON THE RECTANGLE 0 .LT. R .LT. 1 , 0 .LT. Z .LT. 1 WITH THE  
C BOUNDARY CONDITIONS  
C  
C (DU/DR)(1,Z) = 4\*Z\*\*2 FOR 0 .LE. Z .LE. 1  
C  
C AND  
C  
C (DU/DZ)(R,0) = 0 AND (DU/DZ)(R,1) = 4\*R\*\*2 FOR 0 .LE. R .LE. 1 .  
C  
C THE SOLUTION TO THIS PROBLEM IS NOT UNIQUE. IT IS A  
C ONE-PARAMETER FAMILY OF SOLUTIONS GIVEN BY  
C  
C U(R,Z) = (R\*Z)\*\*4 + ARBITRARY CONSTANT .  
C  
C THE R-INTERVAL WILL CONTAIN 50 UNKNOWNS AND THE Z-INTERVAL WILL  
C CONTAIN 52 UNKNOWNS.  
C  
C  
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.  
C  
 IDIMF = 51  
 A = 0.  
 B = 1.  
 M = 50  
 MBDCND = 6  
 C = 0.  
 D = 1.  
 N = 52  
 NBDCND = 3  
 ELMBDA = 0.  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.  
C  
 DO 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  
C GENERATE BOUNDARY DATA.  
C  
 BDB(:N) = 4.\*Z(:N)\*\*4  
C  
C GENERATE BOUNDARY DATA.  
C  
 BDC(:M) = 0.  
 BDD(:M) = 4.\*R(:M)\*\*4  
C  
C BDA IS A DUMMY VARIABLE.  
C  
C 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  
C COMPUTE DISCRETIZATION ERROR BY MINIMIZING OVER ALL A THE FUNCTION  
C NORM(F(I,J) - A\*1 - U(R(I),Z(J))). THE EXACT SOLUTION IS  
C U(R,Z) = (R\*Z)\*\*4 + ARBITRARY CONSTANT.  
C  
 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**

C  
C file thstplr.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THSTPLR  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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, Z  
C-----------------------------------------------  
C  
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.  
C  
 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  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.  
C  
 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  
C GENERATE BOUNDARY DATA.  
C  
 DO J = 1, N  
 BDB(J) = 1. - COS(4.\*THETA(J))  
 END DO  
C  
C GENERATE BOUNDARY DATA.  
C  
 BDC(:M) = 0.  
 BDD(:M) = 0.  
C  
C BDA IS A DUMMY VARIABLE.  
C  
C  
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)  
C  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(R,THETA) = R\*\*4\*(1 - COS(4\*THETA))  
C  
 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 result '  
 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**

C  
C file thstssp.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THSTSSP  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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,ERR,Z  
C-----------------------------------------------  
C  
C 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  
C  
C GENERATE SINES FOR USE IN SUBSEQUENT COMPUTATIONS  
C  
 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  
C  
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  
C  
C STORE DERIVATIVE DATA AT THE EQUATOR  
C  
 BDB(:N) = 0.  
C  
C 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)  
C  
C COMPUTE DISCRETIZATION ERROR. SINCE PROBLEM IS SINGULAR, THE  
C SOLUTION MUST BE NORMALIZED.  
C  
 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-3'  
 WRITE (\*, \*) ' The output from your computer is: '  
 WRITE (\*, \*) ' IERROR =', IERROR, ' PERTRB = ', PERTRB  
 WRITE (\*, \*) ' discretization error = ', ERR  
 STOP   
 END PROGRAM THSTSSP

**THW3CRT**

C  
C file thw3crt.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THW3CRT  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C FROM THE DESCRIPTION OF THE PROBLEM GIVEN ABOVE, WE DEFINE  
C THE FOLLOWING QUANTITIES  
C  
 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  
C  
C FROM THE DIMENSION STATEMENT ABOVE WE DEFINE  
C  
 LDIMF = 11  
 MDIMF = 41  
C  
C WE DEFINE THE GRID POINTS FOR LATER USE.  
C  
 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  
C WE DEFINE THE ARRAY OF DERIVATIVE BOUNDARY VALUES.  
C  
 DO I = 1, LP1  
 DO J = 1, MP1  
 BDZF(I,J) = -X(I)\*\*4\*SIN(Y(J))  
 END DO  
 END DO  
C  
C NOTE THAT FOR THIS EXAMPLE ALL OTHER BOUNDARY ARRAYS ARE  
C DUMMY VARIABLES.  
C 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  
C WE NOW DEFINE THE VALUES OF THE RIGHT SIDE OF THE HELMHOLTZ  
C 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  
C CALL HW3CRT TO GENERATE AND SOLVE THE FINITE DIFFERENCE EQUATION.  
C  
 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)  
C  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION TO THE  
C PROBLEM IS  
C  
C U(X,Y,Z) = X\*\*4\*SIN(Y)\*COS(Z)  
C  
 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 result '  
 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  
C file thwscrt.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THWSCRT  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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,PERTRB,ERR,Z  
C-----------------------------------------------  
C  
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.  
C  
 IDIMF = 45  
 A = 0.  
 B = 2.  
 M = 40  
 MBDCND = 2  
 C = -1.  
 D = 3.  
 N = 80  
 NBDCND = 0  
 ELMBDA = -4.  
C  
C AUXILIARY QUANTITIES.  
C  
 PI = 4.0\*ATAN(1.0)  
 PIBY2 = PI/2.  
 PISQ = PI\*\*2  
 MP1 = M + 1  
 NP1 = N + 1  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE HELMHOLTZ EQUATION.  
C  
 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  
C GENERATE BOUNDARY DATA.  
C  
 DO J = 1, NP1  
 BDB(J) = 4.\*COS((Y(J)+1.)\*PIBY2)  
 END DO  
C  
C BDA, BDC, AND BDD ARE DUMMY VARIABLES.  
C  
 F(1,:NP1) = 0.  
C  
C GENERATE RIGHT SIDE OF EQUATION.  
C  
 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  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C 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**

C  
C file thwscsp.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THWSCSP  
 USE fish  
 implicit none  
 TYPE (fishworkspace) :: w  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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-----------------------------------------------  
C  
C PROGRAM TO ILLUSTRATE THE USE OF HWSCSP  
C  
C  
 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  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING THE  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE EQUATION.  
C  
 MP1 = M + 1  
 DTHETA = TF/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  
C  
C GENERATE NORMAL DERIVATIVE DATA AT EQUATOR  
C  
 BDTF(:NP1) = 0.  
C  
C COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE  
C  
 DO I = 1, MP1  
 F(I,N+1) = COS(THETA(I))\*\*4  
 END DO  
C  
C COMPUTE RIGHT SIDE OF EQUATION  
C  
 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, RS, RF, N,   
 1 NBDCND, BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)  
C  
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  
C THE FOLLOWING PROGRAM ILLUSTRATES THE USE OF HWSCSP TO SOLVE  
C A THREE DIMENSIONAL PROBLEM WHICH HAS LONGITUDNAL DEPENDENCE  
C  
 MBDCND = 2  
 NBDCND = 1  
 DPHI = PI/72.  
 ELMBDA = -2.\*(1. - COS(DPHI))/DPHI\*\*2  
C  
C COMPUTE BOUNDARY DATA ON THE SURFACE OF THE SPHERE  
C  
 DO I = 1, MP1  
 F(I,N+1) = SIN(THETA(I))  
 END DO  
C  
C COMPUTE RIGHT SIDE OF THE EQUATION  
C  
 F(:MP1,:N) = 0.  
 CALL HWSCSP (INTL, TS, TF, M, MBDCND, BDTS, BDTF, RS, RF, N,   
 1 NBDCND, BDRS, BDRF, ELMBDA, F, IDIMF, PERTRB, IERROR, W)  
C  
C COMPUTE DISCRETIZATION ERROR (FOURIER COEFFICIENTS)  
C  
 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**

C  
C file thwscyl.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THWSCYL  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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  
C-----------------------------------------------  
C  
C 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.  
C  
C AUXILIARY QUANTITIES.  
C  
 MP1 = M + 1  
 NP1 = N + 1  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.  
C  
 DO I = 1, MP1  
 R(I) = FLOAT(I - 1)/50.  
 END DO  
 DO J = 1, NP1  
 Z(J) = FLOAT(J - 1)/100.  
 END DO  
C  
C GENERATE BOUNDARY DATA.  
C  
 BDB(:NP1) = 4.\*Z(:NP1)\*\*4  
C  
C GENERATE BOUNDARY DATA.  
C  
 BDC(:MP1) = 0.  
 BDD(:MP1) = 4.\*R(:MP1)\*\*4  
C  
C BDA IS A DUMMY VARIABLE.  
C  
C  
C GENERATE RIGHT SIDE OF EQUATION.  
C  
 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  
C COMPUTE DISCRETIZATION ERROR BY MINIMIZING OVER ALL A THE FUNCTION  
C NORM(F(I,J) - A\*1 - U(R(I),Z(J))). THE EXACT SOLUTION IS  
C U(R,Z) = (R\*Z)\*\*4 + ARBITRARY CONSTANT.  
C  
 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**

C  
C file thwsplr.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM THWSPLR  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
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  
C-----------------------------------------------  
C  
C PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE HWSPLR TO SOLVE  
C THE EQUATION  
C  
C (1/R)(D/DR)(R\*(DU/DR)) + (1/R\*\*2)(D/DTHETA)(DU/DTHETA) = 16\*R\*\*2  
C  
C ON THE QUARTER-DISK 0 .LT. R .LT. 1, 0 .LT. THETA .LT. PI/2 WITH  
C WITH THE BOUNDARY CONDITIONS  
C  
C U(1,THETA) = 1 - COS(4\*THETA), 0 .LE. THETA .LE. 1  
C  
C AND  
C  
C (DU/DTHETA)(R,0) = (DU/DTHETA)(R,PI/2) = 0, 0 .LE. R .LE. 1.  
C  
C (NOTE THAT THE SOLUTION U IS UNSPECIFIED AT R = 0.)  
C THE R-INTERVAL WILL BE DIVIDED INTO 50 PANELS AND THE  
C THETA-INTERVAL WILL BE DIVIDED INTO 48 PANELS.  
C  
C  
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF.  
C  
 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  
C AUXILIARY QUANTITIES.  
C  
 MP1 = M + 1  
 NP1 = N + 1  
C  
C GENERATE AND STORE GRID POINTS FOR THE PURPOSE OF COMPUTING  
C BOUNDARY DATA AND THE RIGHT SIDE OF THE POISSON EQUATION.  
C  
 DO 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  
C GENERATE BOUNDARY DATA.  
C  
 BDC(:MP1) = 0.  
 BDD(:MP1) = 0.  
C  
C BDA AND BDB ARE DUMMY VARIABLES.  
C  
 DO J = 1, NP1  
 F(MP1,J) = 1. - COS(4.\*THETA(J))  
 END DO  
C  
C GENERATE RIGHT SIDE OF EQUATION.  
C  
 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)  
C  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C U(R,THETA) = R\*\*4\*(1 - COS(4\*THETA))  
C  
 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**

C  
C file thwsssp.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
C PROGRAM TO ILLUSTRATE THE USE OF HWSSSP  
C  
 PROGRAM THWSSSP  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C-----------------------------------------------  
 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  
C  
C GENERATE SINES FOR USE IN SUBSEQUENT COMPUTATIONS  
C  
 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  
C  
 DO J = 1, NP1  
 F(:MP1,J) = 2. - 6.\*(SINT(:MP1)\*SINP(J))\*\*2  
 END DO  
C  
C STORE DERIVATIVE DATA AT THE EQUATOR  
C  
 BDTF(:NP1) = 0.  
C  
 CALL HWSSSP (TS, TF, M, MBDCND, BDTS, BDTF, PS, PF, N, NBDCND,   
 1 BDPS, BDPF, ELMBDA, F, IDIMF, PERTRB, IERROR)  
C  
C COMPUTE DISCRETIZATION ERROR. SINCE PROBLEM IS SINGULAR, THE  
C SOLUTION MUST BE NORMALIZED.  
C  
 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  
C  
 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**

C  
C file tpois3d.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TPOIS3D  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 INTEGER::LDIMF,MDIMF,LPEROD,L,MPEROD,M,NPEROD,N,I,J,K,IERROR  
 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  
C-----------------------------------------------  
C  
C FROM THE DIMENSION STATEMENT WE GET THAT LDIMF = 32, MDIMF = 33,  
C  
 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  
C GENERATE GRID POINTS FOR LATER USE.  
C  
 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  
C  
C GENERATE COEFFICIENTS  
C  
 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  
C  
C GENERATE RIGHT SIDE OF EQUATION  
C  
 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  
C CALL POIS3D TO SOLVE EQUATIONS.  
C  
 CALL POIS3D (LPEROD, L, C1, MPEROD, M, C2, NPEROD, N, A, B, C,   
 1 LDIMF, MDIMF, F, IERROR)  
C  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C  
C U(X,Y,Z) = SIN(X)\*SIN(Y)\*(1+Z)\*\*4  
C  
 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**

C  
C file tpoistg.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TPOISTG  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C-----------------------------------------------  
C  
C PROGRAM TO ILLUSTRATE THE USE OF SUBROUTINE POISTG TO  
C SOLVE THE EQUATION  
C  
C (1/COS(X))(D/DX)(COS(X)(DU/DX)) + (D/DY)(DU/DY) =  
C  
C 2\*Y\*\*2\*(6-Y\*\*2)\*SIN(X)  
C  
C ON THE RECTANGLE -PI/2 .LT. X .LT. PI/2 AND  
C 0 .LT. Y .LT. 1 WITH THE BOUNDARY CONDITIONS  
C  
C (DU/DX) (-PI/2,Y) = (DU/DX)(PI/2,Y) = 0 , 0 .LE. Y .LE. 1 (2)  
C  
C U(X,0) = 0 (3)  
C -PI/2 .LE. X .LE. PI/2  
C (DU/DY)(X,1) = 4SIN(X) (4)  
C  
C USING FINITE DIFFERENCES ON A STAGGERED GRID WITH  
C DELTAX (= DX) = PI/40 AND DELTAY (= DY) = 1/20 .  
C TO SET UP THE FINITE DIFFERENCE EQUATIONS WE DEFINE  
C THE GRID POINTS  
C  
C X(I) = -PI/2 + (I-0.5)DX I=1,2,...,40  
C  
C Y(J) = (J-O.5)DY J=1,2,...,20  
C  
C AND LET V(I,J) BE AN APPROXIMATION TO U(X(I),Y(J)).  
C NUMBERING THE GRID POINTS IN THIS FASHION GIVES THE SET  
C OF UNKNOWNS AS V(I,J) FOR I=1,2,...,40 AND J=1,2,...,20.  
C HENCE, IN THE PROGRAM M = 40 AND N = 20. AT THE INTERIOR  
C GRID POINT (X(I),Y(J)), WE REPLACE ALL DERIVATIVES IN  
C EQUATION (1) BY SECOND ORDER CENTRAL FINITE DIFFERENCES,  
C MULTIPLY BY DY\*\*2, AND COLLECT COEFFICIENTS OF V(I,J) TO  
C GET THE FINITE DIFFERENCE EQUATION  
C  
C A(I)V(I-1,J) + B(I)V(I,J) + C(I)V(I+1,J)  
C  
C + V(I,J-1) - 2V(I,J) + V(I,J+1) = F(I,J) (5)  
C  
C WHERE S = (DY/DX)\*\*2, AND FOR I=2,3,...,39  
C  
C A(I) = S\*COS(X(I)-DX/2)  
C  
C B(I) = -S\*(COS(X(I)-DX/2)+COS(X(I)+DX/2))  
C  
C C(I) = S\*COS(X(I)+DX/2)  
C  
C F(I,J) = 2DY\*\*2\*Y(J)\*\*2\*(6-Y(J)\*\*2)\*SIN(X(I)) , J=1,2,...,19.  
C  
C TO OBTAIN EQUATIONS FOR I = 1, WE REPLACE EQUATION (2)  
C BY THE SECOND ORDER APPROXIMATION  
C  
C (V(1,J)-V(0,J))/DX = 0  
C  
C AND USE THIS EQUATION TO ELIMINATE V(0,J) IN EQUATION (5)  
C TO ARRIVE AT THE EQUATION  
C  
C B(1)V(1,J) + C(1)V(2,J) + V(1,J-1) - 2V(1,J) + V(1,J+1)  
C  
C = F(1,J)  
C  
C WHERE  
C  
C B(1) = -S\*(COS(X(1)-DX/2)+COS(X(1)+DX/2))  
C  
C C(1) = -B(1)  
C  
C FOR COMPLETENESS, WE SET A(1) = 0.  
C TO OBTAIN EQUATIONS FOR I = 40, WE REPLACE THE DERIVATIVE  
C IN EQUATION (2) AT X=PI/2 IN A SIMILAR FASHION, USE THIS  
C EQUATION TO ELIMINATE THE VIRTUAL UNKNOWN V(41,J) IN EQUATION  
C (5) AND ARRIVE AT THE EQUATION  
C  
C A(40)V(39,J) + B(40)V(40,J)  
C  
C + V(40,J-1) - 2V(40,J) + V(40,J+1) = F(40,J)  
C  
C WHERE  
C  
C A(40) = -B(40) = -S\*(COS(X(40)-DX/2)+COS(X(40)+DX/2))  
C  
C FOR COMPLETENESS, WE SET C(40) = 0. HENCE, IN THE  
C PROGRAM MPEROD = 1.  
C FOR J = 1, WE REPLACE EQUATION (3) BY THE SECOND ORDER  
C APPROXIMATION  
C  
C (V(I,0) + V(I,1))/2 = 0  
C  
C TO ARRIVE AT THE CONDITION  
C  
C V(I,0) = -V(I,1) .  
C  
C FOR J = 20, WE REPLACE EQUATION (4) BY THE SECOND ORDER  
C APPROXIMATION  
C  
C (V(I,21) - V(I,20))/DY = 4\*SIN(X)  
C  
C AND COMBINE THIS EQUATION WITH EQUATION (5) TO ARRIVE AT  
C THE EQUATION  
C  
C A(I)V(I-1,20) + B(I)V(I,20) + C(I)V(I+1,20)  
C  
C + V(I,19) - 2V(I,20) + V(I,21) = F(I,20)  
C  
C WHERE  
C  
C V(I,21) = V(I,20) AND  
C  
C F(I,20) = 2\*DY\*\*2\*Y(J)\*\*2\*(6-Y(J)\*\*2)\*SIN(X(I)) - 4\*DY\*SIN(X(I))  
C  
C HENCE, IN THE PROGRAM NPEROD = 2 .  
C THE EXACT SOLUTION TO THIS PROBLEM IS  
C  
C U(X,Y) = Y\*\*4\*COS(X) .  
C  
C  
C FROM DIMENSION STATEMENT WE GET VALUE OF IDIMF = 42  
C  
 IDIMF = 42  
 MPEROD = 1  
 M = 40  
 PI = 4.0\*ATAN(1.0)  
 DX = PI/FLOAT(M)  
 NPEROD = 2  
 N = 20  
 DY = 1./FLOAT(N)  
C  
C GENERATE AND STORE GRID POINTS FOR COMPUTATION.  
C  
 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  
C  
C GENERATE COEFFICIENTS .  
C  
 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  
C GENERATE RIGHT SIDE OF EQUATION.  
C  
 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)  
C  
C COMPUTE DISCRETIZATION ERROR. THE EXACT SOLUTION IS  
C  
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**

C  
C file tsepeli.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TSEPELI  
 USE fish  
 implicit none  
 TYPE (fishworkspace) :: w  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 INTEGER::M,N,NX,NY,I,J,MBDCND,NBDCND,IDMN,INTL,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, DF, EF, FF, ALPHA  
 1 , BETA, DUM, PERTRB, ERR, ERR2, ERR4  
C-----------------------------------------------  
C  
C DECLARE COEFFICIENT SUBROUTINES EXTERNAL  
C  
 external cofx,cofy  
C  
C DEFINE ARITHMETIC FUNCTIONS GIVING EXACT SOLUTION  
C  
C  
C SET LIMITS ON REGION  
C  
 A = 0.0  
 B = 1.0  
 C = 0.0  
 D = 1.0  
C  
C SET GRID SIZE  
C  
 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  
C  
C SET SPECIFIED BOUNDARY CONDITIONS AT Y=C,D  
C  
 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  
C SET RIGHT HAND SIDE  
C  
 GRHS(I,J) = AF\*UXXE(X,Y) + BF\*UXE(X,Y) + CF\*UE(X,Y) + DF\*  
 1 UYYE(X,Y) + EF\*UYE(X,Y) + FF\*UE(X,Y)  
 END DO  
 END DO  
C  
C 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  
C  
C SET BOUNDARY SWITHCES  
C  
 MBDCND = 3  
 NBDCND = 1  
C  
C SET FIRST DIMENSION OF USOL,GRHS  
C  
 IDMN = 33  
! set for initialization of sepeli  
 INTL = 0  
C  
C OBTAIN SECOND ORDER APPROXIMATION  
C  
 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  
C  
C OBTAIN FOURTH ORDER APPROXIMATION  
C  
 IORDER = 4  
C  
C 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)  
C  
C COMPUTE DISCRETIZATION ERROR  
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  
! 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'  
 WRITE (\*, \*) ' The output from your computer 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 D u m m y A r g u m e n t s  
C-----------------------------------------------  
 REAL , INTENT(IN) :: X  
 REAL , INTENT(OUT) :: AF  
 REAL , INTENT(OUT) :: BF  
 REAL , INTENT(OUT) :: CF  
C-----------------------------------------------  
C  
C SET COEFFICIENTS IN THE X-DIRECTION.  
C  
 AF = (X + 1.)\*\*2  
 BF = 2.0\*(X + 1.)  
 CF = -X  
 RETURN   
 END SUBROUTINE COFX  
  
  
 SUBROUTINE COFY(Y, DF, EF, FF)  
C-----------------------------------------------  
C D u m m y A r g u m e n t s  
C-----------------------------------------------  
 REAL , INTENT(IN) :: Y  
 REAL , INTENT(OUT) :: DF  
 REAL , INTENT(OUT) :: EF  
 REAL , INTENT(OUT) :: FF  
C-----------------------------------------------  
C  
C SET COEFFICIENTS IN Y DIRECTION  
C  
 DF = EXP(Y)  
 EF = 0.0  
 FF = -Y  
 RETURN   
 END SUBROUTINE COFY

**TSEPX4**

C  
C file tsepx4.f  
C  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C \* \*  
C \* copyright (c) 2005 by UCAR \*  
C \* \*  
C \* University Corporation for Atmospheric Research \*  
C \* \*  
C \* all rights reserved \*  
C \* \*  
C \* FISHPACK90 version 1.1 \*  
C \* \*  
C \* A Package of Fortran 77 and 90 \*  
C \* \*  
C \* Subroutines and Example Programs \*  
C \* \*  
C \* for Modeling Geophysical Processes \*  
C \* \*  
C \* by \*  
C \* \*  
C \* John Adams, Paul Swarztrauber and Roland Sweet \*  
C \* \*  
C \* of \*  
C \* \*  
C \* the National Center for Atmospheric Research \*  
C \* \*  
C \* Boulder, Colorado (80307) U.S.A. \*  
C \* \*  
C \* which is sponsored by \*  
C \* \*  
C \* the National Science Foundation \*  
C \* \*  
C \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*  
C  
 PROGRAM TSEPX4  
 implicit none  
C-----------------------------------------------  
C L o c a l V a r i a b l e s  
C-----------------------------------------------  
 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  
C-----------------------------------------------  
 EXTERNAL COFX4  
C  
C DEFINE ARITHMETIC FUNCTIONS GIVING EXACT SOLUTION  
C  
C  
C SET LIMITS ON REGION  
C  
 A = 0.0  
 B = 1.0  
 C = 0.0  
 D = 1.0  
C  
C SET GRID SIZE  
C  
 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  
C  
C 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  
C  
C SET RIGHT HAND SIDE  
C  
 GRHS(I,J)=AF\*UXXE(X,Y)+BF\*UXE(X,Y)+CF\*UE(X,Y)+UYYE(X,Y)  
 END DO  
 END DO  
C  
C 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  
C  
C SET BOUNDARY SWITHCES  
C  
 MBDCND = 3  
 NBDCND = 1  
C  
C SET FIRST DIMENSION OF USOL,GRHS AND WORK SPACE LENGTH  
C  
 IDMN = 33  
C  
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)  
C  
C COMPUTE SECOND ORDER DISCRETIZATION ERROR (RELATIVE)  
C ALSO RESET SPECIFIED BOUNDARIES AND RIGHT HAND SIDE.  
C  
 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)))  
C  
C RESET RIGHT HAND SIDE IN GRHS FOR FOURTH ORDER APPROXIMATION CALL  
C  
 GRHS(I,J)=AF\*UXXE(X,Y)+BF\*UXE(X,Y)+CF\*UE(X,Y)+UYYE(X,Y)  
 END DO  
 END DO  
 ERR2 = ERR  
C  
C 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  
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 (\*, \*) ' IERROR = 0'  
 WRITE (\*, \*) ' Second Order Discretization 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'  
 WRITE (\*, \*) ' The output from your computer 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-----------------------------------------------  
C D u m m y A r g u m e n t s  
C-----------------------------------------------  
 REAL , INTENT(IN) :: X  
 REAL , INTENT(OUT) :: AF  
 REAL , INTENT(OUT) :: BF  
 REAL , INTENT(OUT) :: CF  
C-----------------------------------------------  
C  
C SET COEFFICIENTS IN THE X-DIRECTION.  
C  
 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