NAG Toolbox

nag_pde_3d_ellip_helmholtz (d03fa)

1 Purpose

nag_pde_3d_ellip_helmholtz (d03fa) solves the Helmholtz equation in Cartesian coordinates in three dimensions using the standard seven-point finite difference approximation. This function is designed to be particularly efficient on vector processors.

2 Syntax

```
[f, pertrb, ifail] = nag_pde_3d_ellip_helmholtz(xs, xf, l, lbdcnd, bdxs, bdxf,
ys, yf, m, mbdcnd, bdys, bdyf, zs, zf, n, nbdcnd, bdzs, bdzf, lambda, f, 'lwrk',
lwrk)
```

```
[f, pertrb, ifail] = d03fa(xs, xf, l, lbdcnd, bdxs, bdxf, ys, yf, m, mbdcnd,
bdys, bdyf, zs, zf, n, nbdcnd, bdzs, bdzf, lambda, f, 'lwrk', lwrk)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: **lwrk** was added as an optional parameter.

3 Description

nag_pde_3d_ellip_helmholtz (d03fa) solves the three-dimensional Helmholtz equation in Cartesian coordinates:

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

This function forms the system of linear equations resulting from the standard seven-point finite difference equations, and then solves the system using a method based on the fast Fourier transform (FFT) described by Swarztrauber (1984). This function is based on the function HW3CRT from FISHPACK (see Swarztrauber and Sweet (1979)).

More precisely, the function replaces all the second derivatives by second-order central difference approximations, resulting in a block tridiagonal system of linear equations. The equations are modified to allow for the prescribed boundary conditions. Either the solution or the derivative of the solution may be specified on any of the boundaries, or the solution may be specified to be periodic in any of the three dimensions. By taking the discrete Fourier transform in the x- and y-directions, the equations are reduced to sets of tridiagonal systems of equations. The Fourier transforms required are computed using the multiple FFT functions found in Chapter C06.

4 References

Swarztrauber P N (1984) Fast Poisson solvers *Studies in Numerical Analysis* (ed G H Golub) Mathematical Association of America

Swarztrauber P N and Sweet R A (1979) Efficient Fortran subprograms for the solution of separable elliptic partial differential equations *ACM Trans. Math. Software* **5** 352–364

5 Parameters

5.1 Compulsory Input Parameters

1: **xs** – REAL (KIND=nag_wp)

The lower bound of the range of x, i.e., $\mathbf{xs} \leq x \leq \mathbf{xf}$.

Constraint: $\mathbf{xs} < \mathbf{xf}$.

2: **xf** – REAL (KIND=nag_wp)

The upper bound of the range of x, i.e., $\mathbf{xs} \leq x \leq \mathbf{xf}$.

Constraint: $\mathbf{xs} < \mathbf{xf}$.

3: **l** – INTEGER

The number of panels into which the interval (**xs**,**xf**) is subdivided. Hence, there will be $\mathbf{l} + 1$ grid points in the x-direction given by $x_i = \mathbf{xs} + (i-1) \times \delta x$, for $i = 1, 2, \ldots, \mathbf{l} + 1$, where $\delta x = (\mathbf{xf} - \mathbf{xs})/\mathbf{l}$ is the panel width.

Constraint: $\mathbf{l} \geq 5$.

4: **lbdcnd** – INTEGER

Indicates the type of boundary conditions at $x = \mathbf{xs}$ and $x = \mathbf{xf}$.

$\mathbf{lbdcnd} = 0$
: If the solution is periodic in x, i.e., $u(\mathbf{xs}, y, z) = u(\mathbf{xf}, y, z)$.

$\mathbf{lbdcnd} = 1$
: If the solution is specified at $x = \mathbf{xs}$ and $x = \mathbf{xf}$.

$\mathbf{lbdcnd} = 2$
: If the solution is specified at $x = \mathbf{xs}$ and the derivative of the solution with respect to x is specified at $x = \mathbf{xf}$.

$\mathbf{lbdcnd} = 3$
: If the derivative of the solution with respect to x is specified at $x = \mathbf{xs}$ and $x = \mathbf{xf}$.

$\mathbf{lbdcnd} = 4$
: If the derivative of the solution with respect to x is specified at $x = \mathbf{xs}$ and the solution is specified at $x = \mathbf{xf}$.

Constraint: $0 \leq \mathbf{lbdcnd} \leq 4$.

5: $\mathbf{bdxs}(\mathit{ldf2}, \mathbf{n} + \mathbf{1})$ – REAL (KIND=nag_wp) array

ldf2, the first dimension of the array, must satisfy the constraint $\mathit{ldf2} \geq \mathbf{m} + 1$.

The values of the derivative of the solution with respect to x at $x = \mathbf{xs}$. When $\mathbf{lbdcnd} = 3$ or 4, $\mathbf{bdxs}(j, k) = u_x(\mathbf{xs}, y_j, z_k)$, for $j = 1, 2, \ldots, \mathbf{m} + 1$ and $k = 1, 2, \ldots, \mathbf{n} + 1$.

When **lbdcnd** has any other value, **bdxs** is not referenced.

6: $\mathbf{bdxf}(\mathit{ldf2}, \mathbf{n} + \mathbf{1})$ – REAL (KIND=nag_wp) array

ldf2, the first dimension of the array, must satisfy the constraint $\mathit{ldf2} \geq \mathbf{m} + 1$.

The values of the derivative of the solution with respect to x at $x = \mathbf{xf}$. When $\mathbf{lbdcnd} = 2$ or 3, $\mathbf{bdxf}(j, k) = u_x(\mathbf{xf}, y_j, z_k)$, for $j = 1, 2, \ldots, \mathbf{m} + 1$ and $k = 1, 2, \ldots, \mathbf{n} + 1$.

When **lbdcnd** has any other value, **bdxf** is not referenced.

7: **ys** – REAL (KIND=nag_wp)

The lower bound of the range of y, i.e., $\mathbf{ys} \le y \le \mathbf{yf}$.

Constraint: $\mathbf{ys} < \mathbf{yf}$.

8: **yf** – REAL (KIND=nag_wp)

The upper bound of the range of y, i.e., $\mathbf{ys} \le y \le \mathbf{yf}$.

Constraint: $\mathbf{ys} < \mathbf{yf}$.

9: **m** – INTEGER

The number of panels into which the interval (**ys**,**yf**) is subdivided. Hence, there will be $\mathbf{m} + 1$ grid points in the y-direction given by $y_j = \mathbf{ys} + (j - 1) \times \delta y$, for $j = 1, 2, \ldots, \mathbf{m} + 1$, where $\delta y = (\mathbf{yf} - \mathbf{ys})/\mathbf{m}$ is the panel width.

Constraint: $\mathbf{m} \ge 5$.

10: **mbdcnd** – INTEGER

Indicates the type of boundary conditions at $y = \mathbf{ys}$ and $y = \mathbf{yf}$.

$\mathbf{mbdcnd} = 0$
: If the solution is periodic in y, i.e., $u(x, \mathbf{yf}, z) = u(x, \mathbf{ys}, z)$.

$\mathbf{mbdcnd} = 1$
: If the solution is specified at $y = \mathbf{ys}$ and $y = \mathbf{yf}$.

$\mathbf{mbdcnd} = 2$
: If the solution is specified at $y = \mathbf{ys}$ and the derivative of the solution with respect to y is specified at $y = \mathbf{yf}$.

$\mathbf{mbdcnd} = 3$
: If the derivative of the solution with respect to y is specified at $y = \mathbf{ys}$ and $y = \mathbf{yf}$.

$\mathbf{mbdcnd} = 4$
: If the derivative of the solution with respect to y is specified at $y = \mathbf{ys}$ and the solution is specified at $y = \mathbf{yf}$.

Constraint: $0 \le \mathbf{mbdcnd} \le 4$.

11: $\mathbf{bdys}(ldf, \mathbf{n+1})$ – REAL (KIND=nag_wp) array

ldf, the first dimension of the array, must satisfy the constraint $ldf \ge \mathbf{l} + 1$.

The values of the derivative of the solution with respect to y at $y = \mathbf{ys}$. When $\mathbf{mbdcnd} = 3$ or 4, $\mathbf{bdys}(i, k) = u_y(x_i, \mathbf{ys}, z_k)$, for $i = 1, 2, \ldots, \mathbf{l} + 1$ and $k = 1, 2, \ldots, \mathbf{n} + 1$.

When **mbdcnd** has any other value, **bdys** is not referenced.

12: $\mathbf{bdyf}(ldf, \mathbf{n+1})$ – REAL (KIND=nag_wp) array

ldf, the first dimension of the array, must satisfy the constraint $ldf \ge \mathbf{l} + 1$.

The values of the derivative of the solution with respect to y at $y = \mathbf{yf}$. When $\mathbf{mbdcnd} = 2$ or 3, $\mathbf{bdyf}(i, k) = u_y(x_i, \mathbf{yf}, z_k)$, for $i = 1, 2, \ldots, \mathbf{l} + 1$ and $k = 1, 2, \ldots, \mathbf{n} + 1$.

When **mbdcnd** has any other value, **bdyf** is not referenced.

13: **zs** – REAL (KIND=nag_wp)

The lower bound of the range of z, i.e., $\mathbf{zs} \le z \le \mathbf{zf}$.

Constraint: $\mathbf{zs} < \mathbf{zf}$.

14: **zf** – REAL (KIND=nag_wp)

The upper bound of the range of z, i.e., $\mathbf{zs} \leq z \leq \mathbf{zf}$.

Constraint: $\mathbf{zs} < \mathbf{zf}$.

15: **n** – INTEGER

The number of panels into which the interval (**zs**,**zf**) is subdivided. Hence, there will be $\mathbf{n} + 1$ grid points in the z-direction given by $z_k = \mathbf{zs} + (k - 1) \times \delta z$, for $k = 1, 2, \ldots, \mathbf{n} + 1$, where $\delta z = (\mathbf{zf} - \mathbf{zs})/\mathbf{n}$ is the panel width.

Constraint: $\mathbf{n} \geq 5$.

16: **nbdcnd** – INTEGER

Specifies the type of boundary conditions at $z = \mathbf{zs}$ and $z = \mathbf{zf}$.

$\mathbf{nbdcnd} = 0$
: if the solution is periodic in z, i.e., $u(x, y, \mathbf{zf}) = u(x, y, \mathbf{zs})$.

$\mathbf{nbdcnd} = 1$
: if the solution is specified at $z = \mathbf{zs}$ and $z = \mathbf{zf}$.

$\mathbf{nbdcnd} = 2$
: if the solution is specified at $z = \mathbf{zs}$ and the derivative of the solution with respect to z is specified at $z = \mathbf{zf}$.

$\mathbf{nbdcnd} = 3$
: if the derivative of the solution with respect to z is specified at $z = \mathbf{zs}$ and $z = \mathbf{zf}$.

$\mathbf{nbdcnd} = 4$
: if the derivative of the solution with respect to z is specified at $z = \mathbf{zs}$ and the solution is specified at $z = \mathbf{zf}$.

Constraint: $0 \leq \mathbf{nbdcnd} \leq 4$.

17: $\mathbf{bdzs}(ldf, \mathbf{m+1})$ – REAL (KIND=nag_wp) array

ldf, the first dimension of the array, must satisfy the constraint $ldf \geq \mathbf{l} + 1$.

The values of the derivative of the solution with respect to z at $z = \mathbf{zs}$. When $\mathbf{nbdcnd} = 3$ or 4, $\mathbf{bdzs}(i, j) = u_z(x_i, y_j, \mathbf{zs})$, for $i = 1, 2, \ldots, \mathbf{l} + 1$ and $j = 1, 2, \ldots, \mathbf{m} + 1$.

When **nbdcnd** has any other value, **bdzs** is not referenced.

18: $\mathbf{bdzf}(ldf, \mathbf{m+1})$ – REAL (KIND=nag_wp) array

ldf, the first dimension of the array, must satisfy the constraint $ldf \geq \mathbf{l} + 1$.

The values of the derivative of the solution with respect to z at $z = \mathbf{zf}$. When $\mathbf{nbdcnd} = 2$ or 3, $\mathbf{bdzf}(i, j) = u_z(x_i, y_j, \mathbf{zf})$, for $i = 1, 2, \ldots, \mathbf{l} + 1$ and $j = 1, 2, \ldots, \mathbf{m} + 1$.

When **nbdcnd** has any other value, **bdzf** is not referenced.

19: **lambda** – REAL (KIND=nag_wp)

The constant λ in the Helmholtz equation. For certain positive values of λ a solution to the differential equation may not exist, and close to these values the solution of the discretized problem will be extremely ill-conditioned. If $\lambda > 0$, then nag_pde_3d_ellip_helmholtz (d03fa) will set $\mathbf{ifail} = 3$, but will still attempt to find a solution. However, since in general the values of λ for which no solution exists cannot be predicted *a priori*, you are advised to treat any results computed with $\lambda > 0$ with great caution.

20: $\mathbf{f}(ldf, ldf2, \mathbf{n+1})$ – REAL (KIND=nag_wp) array

ldf, the first dimension of the array, must satisfy the constraint $ldf \geq \mathbf{l} + 1$.

The values of the right-side of the Helmholtz equation and boundary values (if any).

$$\mathbf{f}(i,j,k) = f(x_i, y_j, z_k) \quad i = 2,3,\ldots,\mathbf{l}, j = 2,3,\ldots,\mathbf{m} \text{ and } k = 2,3,\ldots,\mathbf{n}.$$

On the boundaries **f** is defined by

lbdcnd	$\mathbf{f}(1,j,k)$	$\mathbf{f}(\mathbf{l}+1,j,k)$	
0	$f(\mathbf{xs}, y_j, z_k)$	$f(\mathbf{xs}, y_j, z_k)$	
1	$u(\mathbf{xs}, y_j, z_k)$	$u(\mathbf{xf}, y_j, z_k)$	
2	$u(\mathbf{xs}, y_j, z_k)$	$f(\mathbf{xf}, y_j, z_k)$	$j = 1,2,\ldots,\mathbf{m}+1$
3	$f(\mathbf{xs}, y_j, z_k)$	$f(\mathbf{xf}, y_j, z_k)$	$k = 1,2,\ldots,\mathbf{n}+1$
4	$f(\mathbf{xs}, y_j, z_k)$	$u(\mathbf{xf}, y_j, z_k)$	

mbdcnd	$\mathbf{f}(i,1,k)$	$\mathbf{f}(i,\mathbf{m}+1,k)$	
0	$f(x_i, \mathbf{ys}, z_k)$	$f(x_i, \mathbf{ys}, z_k)$	
1	$u(\mathbf{ys}, x_i, z_k)$	$u(\mathbf{yf}, x_i, z_k)$	
2	$u(x_i, \mathbf{ys}, z_k)$	$f(x_i, \mathbf{yf}, z_k)$	$i = 1,2,\ldots,\mathbf{l}+1$
3	$f(x_i, \mathbf{ys}, z_k)$	$f(x_i, \mathbf{yf}, z_k)$	$k = 1,2,\ldots,\mathbf{n}+1$
4	$f(x_i, \mathbf{ys}, z_k)$	$u(x_i, \mathbf{yf}, z_k)$	

nbdcnd	$\mathbf{f}(i,j,1)$	$\mathbf{f}(i,j,\mathbf{n}+1)$	
0	$f(x_i, y_j, \mathbf{zs})$	$f(x_i, y_j, \mathbf{zs})$	
1	$u(x_i, y_j, \mathbf{zs})$	$u(x_i, y_j, \mathbf{zf})$	
2	$u(x_i, y_j, \mathbf{zs})$	$f(x_i, y_j, \mathbf{zf})$	$i = 1,2,\ldots,\mathbf{l}+1$
3	$f(x_i, y_j, \mathbf{zs})$	$f(x_i, y_j, \mathbf{zf})$	$j = 1,2,\ldots,\mathbf{m}+1$
4	$f(x_i, y_j, \mathbf{zs})$	$u(x_i, y_j, \mathbf{zf})$	

Note: if the table calls for both the solution u and the right-hand side f on a boundary, then the solution must be specified.

5.2 Optional Input Parameters

1: **lwrk** – INTEGER

Default: $2 \times (\mathbf{n}+1) \times \max(\mathbf{l}, \mathbf{m}) + 3 \times \mathbf{l} + 3 \times \mathbf{m} + 4 \times \mathbf{n} + 6$

The dimension of the array **w**. $2 \times (\mathbf{n}+1) \times \max(\mathbf{l}, \mathbf{m}) + 3 \times \mathbf{l} + 3 \times \mathbf{m} + 4 \times \mathbf{n} + 6$ is an upper bound on the required size of w. If **lwrk** is too small, the function exits with **ifail** $= 2$, and if on entry **ifail** $= 0$ or -1, a message is output giving the exact value of **lwrk** required to solve the current problem.

5.3 Output Parameters

1: $\mathbf{f}(ldf, ldf2, \mathbf{n}+1)$ – REAL (KIND=nag_wp) array

$ldf2 = \mathbf{m} + 1$.

Contains the solution $u(i,j,k)$ of the finite difference approximation for the grid point (x_i, y_j, z_k), for $i = 1,2,\ldots,\mathbf{l}+1$, $j = 1,2,\ldots,\mathbf{m}+1$ and $k = 1,2,\ldots,\mathbf{n}+1$.

2: **pertrb** – REAL (KIND=nag_wp)

pertrb $= 0$, unless a solution to Poisson's equation $(\lambda = 0)$ is required with a combination of periodic or derivative boundary conditions (**lbdcnd**, **mbdcnd** and **nbdcnd** $= 0$ or 3). In this case a solution may not exist. **pertrb** is a constant, calculated and subtracted from the array **f**, which ensures that a solution exists. nag_pde_3d_ellip_helmholtz (d03fa) then computes this solution, which is a least squares solution to the original approximation. This solution is not unique and is unnormalized. The value of **pertrb** should be small compared to the right-hand side **f**, otherwise a solution has been obtained to an essentially different problem. This comparison should always be made to ensure that a meaningful solution has been obtained.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **xs** $\geq$ **xf**,
or **l** < 5,
or **lbdcnd** < 0,
or **lbdcnd** > 4,
or **ys** $\geq$ **yf**,
or **m** < 5,
or **mbdcnd** < 0,
or **mbdcnd** > 4,
or **zs** $\geq$ **zf**,
or **n** < 5,
or **nbdcnd** < 0,
or **nbdcnd** > 4,
or $ldf <$ **l** $+ 1$,
or $ldf2 <$ **m** $+ 1$.

ifail $= 2$

On entry, **lwrk** is too small.

ifail $= 3$ (*warning*)

On entry, $\lambda > 0$.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

The execution time is roughly proportional to $\mathbf{l} \times \mathbf{m} \times \mathbf{n} \times (\log_2 \mathbf{l} + \log_2 \mathbf{m} + 5)$, but also depends on input arguments **lbdcnd** and **mbdcnd**.

9 Example

This example solves the Helmholz equation

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

for $(x, y, z) \in [0, 1] \times [0, 2\pi] \times \left[0, \frac{\pi}{2}\right]$, where $\lambda = -2$, and $f(x, y, z)$ is derived from the exact solution

$$u(x,y,z) = x^4 \sin y \cos z.$$

The equation is subject to the following boundary conditions, again derived from the exact solution given above.

$u(0,y,z)$ and $u(1,y,z)$ are prescribed (i.e., **lbdcnd** $= 1$).

$u(x,0,z) = u(x,2\pi,z)$ (i.e., **mbdcnd** $= 0$).

$u(x,y,0)$ and $u_x\left(x,y,\frac{\pi}{2}\right)$ are prescribed (i.e., **nbdcnd** $= 2$).

9.1 Program Text

```
     function d03fa_example

fprintf('d03fa example results\n\n');

xs = 0; xf = 1;    l = 16;
ys = 0; yf = 2*pi; m = 32;
zs = 0; zf = pi/2; n = 20;

bdxs = zeros(m+1, n+1); bdxf = zeros(m+1, n+1);
bdys = zeros(l+1, n+1); bdyf = zeros(l+1, n+1);
bdzs = zeros(l+1, m+1); bdzf = zeros(l+1, m+1);

lbdcnd = nag_int(1);
mbdcnd = nag_int(0);
nbdcnd = nag_int(2);
lambda = -2;

% Define the grid points for later use.
dx = (xf-xs)/l;  x  = [xs:dx:xf];
dy = (yf-ys)/m;  y  = [ys:dy:yf];
dz = (zf-zs)/n;  z  = [zs:dz:zf];

% Define the array of derivative boundary values (z only here).
XY = [x.^4]'*sin(y);
bdzf = -XY;

% Define the f array including boundary conditions.
YZ = [sin(y)]'*cos(z);
for i = 1:l
  f(i,:,:) = (4*x(i)^2*(3-x(i)^2))*YZ;
  g(i,:,:) = x(i)^4*YZ;
end
g(l+1,:,:) = YZ;
f(1,:,:)   = 0;
f(l+1,:,:) = YZ;
f(:,:,1)   = XY;

[u, pertrb, ifail] = d03fa( ...
                            xs, xf, nag_int(l), lbdcnd, bdxs, bdxf, ...
                            ys, yf, nag_int(m), mbdcnd, bdys, bdyf, ...
                            zs, zf, nag_int(n), nbdcnd, bdzs, bdzf, lambda, f);
% calculate error
maxerr   = max(max(max(abs(u - g))));
fprintf('Maximum error in computed solution = %10.3e\n',maxerr);

fig1 = figure;
[xs,ys,zs] = meshgrid(y,x,z);
slice(xs,ys,zs,u,[1.8 5],[0.95],[0.78]);
axis([y(1) y(end) x(1) x(end) 0 2]);
xlabel('y'); ylabel('x'); zlabel('z');
title('Helholtz Equation solution in a box');
view(111, 26);
h = colorbar;
ylabel(h, 'u(x,y,z)')
```

9.2 Program Results

```
     d03fa example results

Maximum error in computed solution =  5.177e-04
```