NAG Toolbox

nag_mesh_2d_renumber (d06cc)

1 Purpose

nag_mesh_2d_renumber (d06cc) renumbers the vertices of a given mesh using a Gibbs method, in order the reduce the bandwidth of Finite Element matrices associated with that mesh.

2 Syntax

```
[nnz, coor, edge, conn, irow, icol, ifail] = nag_mesh_2d_renumber(nnzmax, coor, edge, conn, itrace, 'nv', nv, 'nelt', nelt, 'nedge', nedge)
[nnz, coor, edge, conn, irow, icol, ifail] = d06cc(nnzmax, coor, edge, conn, itrace, 'nv', nv, 'nelt', nelt, 'nedge', nedge)
```

3 Description

nag_mesh_2d_renumber (d06cc) uses a Gibbs method to renumber the vertices of a given mesh in order to reduce the bandwidth of the associated finite element matrix A. This matrix has elements a_{ij} such that:

$$a_{ij} \neq 0 \Rightarrow i \text{ and } j \text{ are vertices belonging to the same triangle.}$$

This function reduces the bandwidth m, which is the smallest integer such that $a_{ij} \neq 0$ whenever $|i-j| > m$ (see Gibbs *et al.* (1976) for details about that method). nag_mesh_2d_renumber (d06cc) also returns the sparsity structure of the matrix associated with the renumbered mesh.

This function is derived from material in the MODULEF package from INRIA (Institut National de Recherche en Informatique et Automatique).

4 References

Gibbs N E, Poole W G Jr and Stockmeyer P K (1976) An algorithm for reducing the bandwidth and profile of a sparse matrix *SIAM J. Numer. Anal.* **13** 236–250

5 Parameters

5.1 Compulsory Input Parameters

1: **nnzmax** – INTEGER

The maximum number of nonzero entries in the matrix based on the input mesh. It is the dimension of the arrays **irow** and **icol** as declared in the function from which nag_mesh_2d_renumber (d06cc) is called.

Constraint: $4 \times \mathbf{nelt} + \mathbf{nv} \leq \mathbf{nnzmax} \leq \mathbf{nv}^2$.

2: **coor**$(\mathbf{2}, \mathbf{nv})$ – REAL (KIND=nag_wp) array

coor$(1, i)$ contains the x coordinate of the ith input mesh vertex, for $i = 1, 2, \ldots, \mathbf{nv}$; while **coor**$(2, i)$ contains the corresponding y coordinate.

3: **edge**$(\mathbf{3}, \mathbf{nedge})$ – INTEGER array

The specification of the boundary or interface edges. **edge**$(1, j)$ and **edge**$(2, j)$ contain the vertex numbers of the two end points of the jth boundary edge. **edge**$(3, j)$ is a user-supplied tag for the

jth boundary or interface edge: $\mathbf{edge}(3,j) = 0$ for an interior edge and has a nonzero tag otherwise.

Constraint: $1 \leq \mathbf{edge}(i,j) \leq \mathbf{nv}$ and $\mathbf{edge}(1,j) \neq \mathbf{edge}(2,j)$, for $i = 1,2$ and $j = 1,2,\ldots,\mathbf{nedge}$.

4: **conn**(**3**, **nelt**) – INTEGER array

The connectivity of the mesh between triangles and vertices. For each triangle j, $\mathbf{conn}(i,j)$ gives the indices of its three vertices (in anticlockwise order), for $i = 1,2,3$ and $j = 1,2,\ldots,\mathbf{nelt}$.

Constraint: $1 \leq \mathbf{conn}(i,j) \leq \mathbf{nv}$ and $\mathbf{conn}(1,j) \neq \mathbf{conn}(2,j)$ and $\mathbf{conn}(1,j) \neq \mathbf{conn}(3,j)$ and $\mathbf{conn}(2,j) \neq \mathbf{conn}(3,j)$, for $i = 1,2,3$ and $j = 1,2,\ldots,\mathbf{nelt}$.

5: **itrace** – INTEGER

The level of trace information required from nag_mesh_2d_renumber (d06cc).

itrace ≤ 0
No output is generated.

itrace $= 1$
Information about the effect of the renumbering on the finite element matrix are output. This information includes the half bandwidth and the sparsity structure of this matrix before and after renumbering.

itrace > 1
The output is similar to that produced when **itrace** $= 1$ but the sparsities (for each row of the matrix, indices of nonzero entries) of the matrix before and after renumbering are also output.

5.2 Optional Input Parameters

1: **nv** – INTEGER

Default: the dimension of the array **coor**.

The total number of vertices in the input mesh.

Constraint: $\mathbf{nv} \geq 3$.

2: **nelt** – INTEGER

Default: the dimension of the array **conn**.

The number of triangles in the input mesh.

Constraint: $\mathbf{nelt} \leq 2 \times \mathbf{nv} - 1$.

3: **nedge** – INTEGER

Default: the dimension of the array **edge**.

The number of boundary edges in the input mesh.

Constraint: $\mathbf{nedge} \geq 1$.

5.3 Output Parameters

1: **nz** – INTEGER

The number of nonzero entries in the matrix based on the input mesh.

2: **coor**(**2**, **nv**) – REAL (KIND=nag_wp) array

$\mathbf{coor}(1,i)$ will contain the x coordinate of the ith renumbered mesh vertex, for $i = 1,2,\ldots,\mathbf{nv}$; while $\mathbf{coor}(2,i)$ will contain the corresponding y coordinate.

3: **edge**$(\mathbf{3}, \mathbf{nedge})$ – INTEGER array

The renumbered specification of the boundary or interface edges.

4: **conn**$(\mathbf{3}, \mathbf{nelt})$ – INTEGER array

The renumbered connectivity of the mesh between triangles and vertices.

5: **irow**$(\mathbf{nnzmax})$ – INTEGER array
6: **icol**$(\mathbf{nnzmax})$ – INTEGER array

The first **nnz** elements contain the row and column indices of the nonzero elements supplied in the finite element matrix A.

7: **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, **nv** < 3,
or **nelt** $> 2 \times$ **nv** $- 1$,
or **nedge** < 1,
or **nnzmax** $< 4 \times$ **nelt** $+$ **nv** or **nnzmax** $>$ **nv**^{2}
or **conn**$(i, j) < 1$ or **conn**$(i, j) >$ **nv** for some $i = 1, 2, 3$ and $j = 1, 2, \ldots,$ **nelt**,
or **conn**$(1, j) =$ **conn**$(2, j)$ or **conn**$(1, j) =$ **conn**$(3, j)$ or
conn$(2, j) =$ **conn**$(3, j)$ for some $j = 1, 2, \ldots,$ **nelt**,
or **edge**$(i, j) < 1$ or **edge**$(i, j) >$ **nv** for some $i = 1, 2$ and $j = 1, 2, \ldots,$ **nedge**,
or **edge**$(1, j) =$ **edge**$(2, j)$ for some $j = 1, 2, \ldots,$ **nedge**,
or $liwork < \max($**nnzmax**$, 20 \times$ **nv**$)$,
or $lrwork <$ **nv**.

ifail $= 2$

A serious error has occurred during the computation of the compact sparsity of the finite element matrix or in an internal call to the renumbering function. Check the input mesh, especially the connectivity between triangles and vertices (the argument **conn**). If the problem persists, contact NAG.

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

None.

9 Example

In this example, a geometry with two holes (two interior circles inside an exterior one) is considered. The geometry has been meshed using the simple incremental method (nag_mesh_2d_gen_inc (d06aa)) and it has 250 vertices and 402 triangles. The function nag_mesh_2d_gen_boundary (d06ba) is used to renumber the vertices, and one can see the benefit in terms of the sparsity of the finite element matrix based on the renumbered mesh.

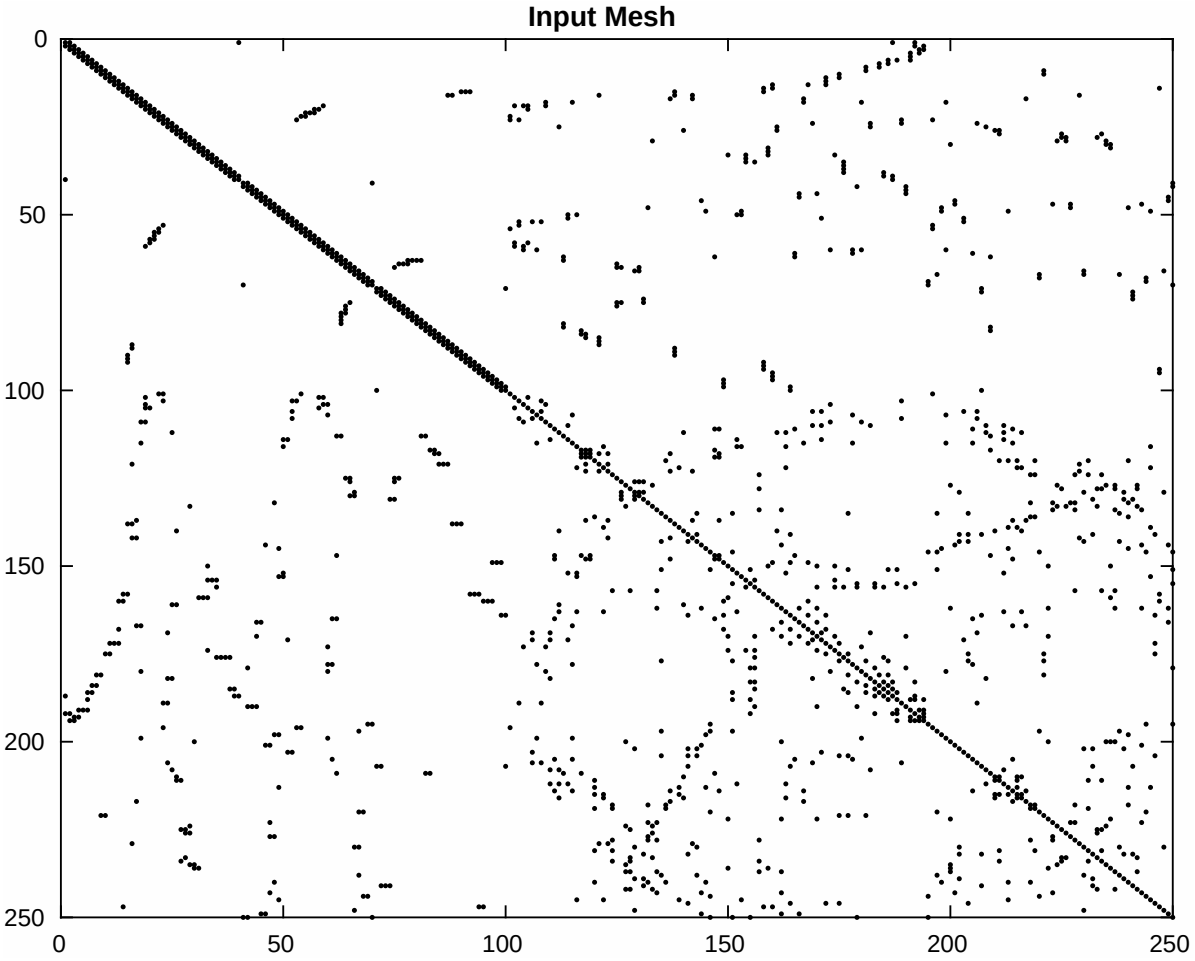
9.1 Program Text

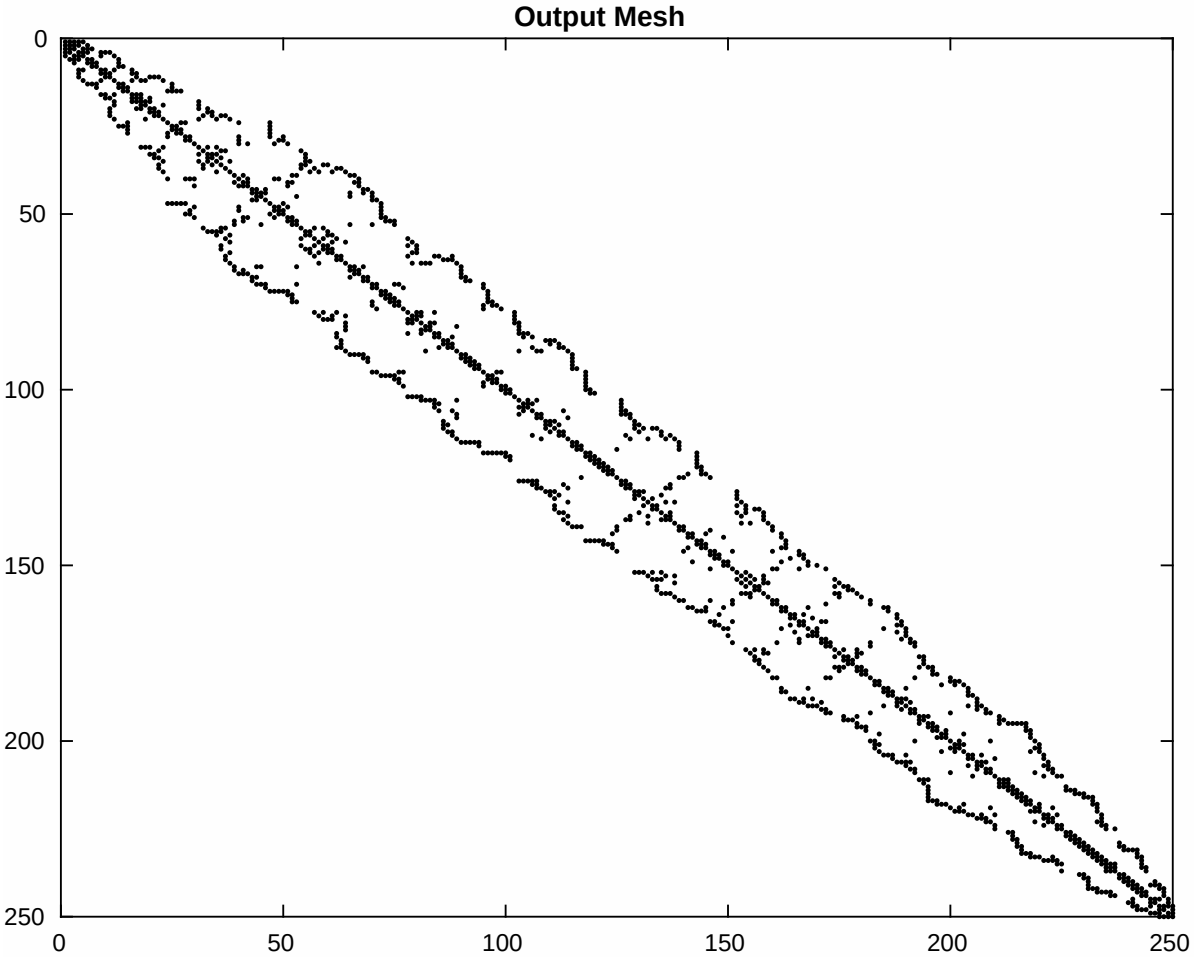
```
     function d06cc_example

fprintf('d06cc example results\n\n');

edge = zeros(3, 100, nag_int_name);
coor = zeros(2, 250);

% Define boundaries
ncirc     = 3; % 3 circles
nvertices = [40, 30, 30];
radii     = [1, 0.49, 0.15];
centres   = [0, 0; -0.5, 0; -0.5, 0.65];

% First circle is outer circle
csign = 1;
i1 = 0;
nvb = 0;
for icirc = 1:ncirc
   for i = 0:nvertices(icirc)-1
      i1 = i1+1;
      theta = 2*pi*i/nvertices(icirc);
      coor(1,i1) = radii(icirc)*cos(theta) + centres(icirc, 1);
      coor(2,i1) = csign*radii(icirc)*sin(theta) +  centres(icirc, 2);
      edge(1,i1) = i1;
      edge(2,i1) = i1 + 1;
      edge(3,i1) = 1;
   end
   edge(2,i1) = nvb + 1;
   nvb = nvb + nvertices(icirc);
   % Subsequent circles are inner circles
   csign = -1;
end
nedge = nvb;

% Initialise mesh control parameters
bspace = zeros(1, 100);
bspace(1:nvb) = 0.05;
smooth = true;
itrace = nag_int(0);

nnzmax = nag_int(3000);

% Mesh geometry
[nv, nelt, coor, conn, ifail] = d06aa(edge, coor, bspace, smooth, itrace);

% Compute the sparsity of the FE matrix from the input geometry
[nz, irow, icol, ifail] = d06cb(nv, nnzmax, conn, 'nelt', nelt);

fprintf('\nNumber of non-zero entries in input mesh:  %d\n', nz);

% Plot sparsity of input mesh
fig1 = figure;
plot(irow(1:double(nz)), icol(1:double(nz)), '.');
title ('Input Mesh');
set(gca,'YDir','reverse');

% Call the renumbering routine and get the new sparsity
[nz, coor, edge, conn, irow, icol, ifail] = ...
    d06cc(nnzmax, coor, edge, conn, itrace, 'nelt', nelt);
```

```
fprintf('Number of non-zero entries in output mesh: %d\n', nz);
% Plot smoothed mesh
fig2 = figure;
plot(irow(1:double(nz)), icol(1:double(nz)), '.');
title ('Output Mesh');
set(gca,'YDir','reverse');
```

9.2 Program Results

```
     d06cc example results

Number of non-zero entries in input mesh:  1556
Number of non-zero entries in output mesh: 1556
```




Output Mesh
0
50
100
150
200
250
0
50
100
150
200
250