NAG Toolbox

nag_mesh_2d_smooth_bary (d06ca)

1 Purpose

nag_mesh_2d_smooth_bary (d06ca) uses a barycentering technique to smooth a given mesh.

2 Syntax

```
[coor, ifail] = nag_mesh_2d_smooth_bary(coor, edge, conn, numfix, itrace, nqint,
'nv', nv, 'nelt', nelt, 'nedge', nedge, 'nvfix', nvfix)
```

```
[coor, ifail] = d06ca(coor, edge, conn, numfix, itrace, nqint, 'nv', nv, 'nelt',
nelt, 'nedge', nedge, 'nvfix', nvfix)
```

3 Description

nag_mesh_2d_smooth_bary (d06ca) uses a barycentering approach to improve the smoothness of a given mesh. The measure of quality used for a triangle K is

$$Q_K = \alpha \frac{h_K}{\rho_K};$$

where h_K is the diameter (length of the longest edge) of K, ρ_K is the radius of its inscribed circle and $\alpha = \frac{\sqrt{3}}{6}$ is a normalization factor chosen to give $Q_K = 1$ for an equilateral triangle. Q_K ranges from 1, for an equilateral triangle, to ∞, for a totally flat triangle.

nag_mesh_2d_smooth_bary (d06ca) makes small perturbation to vertices (using a barycenter formula) in order to give a reasonably good value of Q_K for all neighbouring triangles. Some vertices may optionally be excluded from this process.

For more details about the smoothing method, especially with regard to differing quality, consult the D06 Chapter Introduction as well as George and Borouchaki (1998).

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

4 References

George P L and Borouchaki H (1998) *Delaunay Triangulation and Meshing: Application to Finite Elements* Editions HERMES, Paris

5 Parameters

5.1 Compulsory Input Parameters

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

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

2: **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: **edge**$(3, j) = 0$ for an interior edge and has a nonzero tag otherwise.

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

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

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

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

4: **numfix**(:) – INTEGER array

The dimension of the array **numfix** must be at least max(1, **nvfix**)

The indices in **coor** of fixed interior vertices of the input mesh.

Constraint: if **nvfix** > 0, $1 \leq$ **numfix**$(i) \leq$ **nv**, for $i = 1, 2, \ldots,$ **nvfix**.

5: **itrace** – INTEGER

The level of trace information required from nag_mesh_2d_smooth_bary (d06ca).

itrace ≤ 0
: No output is generated.

itrace $= 1$
: A histogram of the triangular element qualities is printed on the current advisory message unit (see nag_file_set_unit_advisory (x04ab)) before and after smoothing. This histogram gives the lowest and the highest triangle quality as well as the number of elements lying in each of the **nqint** equal intervals between the extremes.

itrace > 1
: The output is similar to that produced when **itrace** $= 1$ but the connectivity between vertices and triangles (for each vertex, the list of triangles in which it appears) is given.

You are advised to set **itrace** $= 0$, unless you are experienced with finite element meshes.

6: **nqint** – INTEGER

The number of intervals between the extreme quality values for the input and the smoothed mesh.

If **itrace** $= 0$, **nqint** is not referenced.

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: **nv** ≥ 3.

2: **nelt** – INTEGER

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

The number of triangles in the input mesh.

Constraint: **nelt** $\leq 2 \times$ **nv** $- 1$.

3: **nedge** – INTEGER

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

The number of the boundary and interface edges in the input mesh.

Constraint: **nedge** ≥ 1.

4: **nvfix** – INTEGER

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

The number of fixed vertices in the input mesh.

Constraint: $0 \leq$ **nvfix** $\leq$ **nv**.

5.3 Output Parameters

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

coor$(1, i)$ will contain the x coordinate of the ith smoothed mesh vertex, for $i = 1, 2, \ldots,$ **nv**; while **coor**$(2, i)$ will contain the corresponding y coordinate. Note that the coordinates of boundary and interface edge vertices, as well as those specified by you (see the description of **numfix**), are unchanged by the process.

2: **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 **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 **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 **nvfix** < 0 or **nvfix** $>$ **nv**,
or **numfix**$(i) < 1$ or **numfix**$(i) >$ **nv** for some $i = 1, 2, \ldots,$ **nvfix** if **nvfix** > 0,
or $liwork < 8 \times$ **nelt** $+ 2 \times$ **nv**,
or $lrwork < 2 \times$ **nv** $+$ **nelt**.

ifail $= 2$

A serious error has occurred in an internal call to an auxiliary function. Check the input mesh, especially the connectivity between triangles and vertices (the argument **conn**). Setting **itrace** > 1 may provide more information. 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 uniform mesh on the unit square is randomly distorted using functions from Chapter G05. nag_mesh_2d_smooth_bary (d06ca) is then used to smooth the distorted mesh and recover a uniform mesh.

9.1 Program Text

```
     function d06ca_example

fprintf('d06ca example results\n\n');

imax = 20;
jmax = 20;
delta = 87;
nv = imax*jmax;

hx = 1/(imax-1);
hy = 1/(jmax-1);
rad = 0.01*delta*min(hx,hy)/2;

% Initialise the seed for the random number generator
seed = [nag_int(1762541)];
% genid and subid identify the base generator
genid = nag_int(1);
subid = nag_int(1);
% Initialise the generator to a repeatable sequence
[state, ifail] = g05kf(genid, subid, seed);

% Generate two sets of uniform random variates
[state, x1, ifail] = g05sq(nag_int(nv), 0, rad, state);
[state, x2, ifail] = g05sq(nag_int(nv), 0, 2*pi, state);

% Generate a simple uniform mesh and then distort it
% randomly within the distortion neighbourhood of each node.
coor = zeros(2, nv);
conn = zeros(3, 2*nv-1, nag_int_name);
k = 0;
ind = 0;
for j = 1:jmax
  for i = 1:imax
    k = k + 1;
    r = x1(k);
    theta = x2(k);

    if (i==1 || i==imax || j==1 || j==jmax)
      r = 0;
    end

    coor(1,k) = (i-1)*hx + r*cos(theta);
    coor(2,k) = (j-1)*hy + r*sin(theta);

    if (i<imax && j<jmax)
      ind = ind + 1;
      conn(1,ind) = k;
      conn(2,ind) = k + 1;
      conn(3,ind) = k + imax + 1;
      ind = ind + 1;
      conn(1,ind) = k;
```

```
      conn(2,ind) = k + imax + 1;
      conn(3,ind) = k + imax;
    end
  end
end

nelt = ind;

% Boundary edges
nedge = 0;
edge = zeros(3, 100, nag_int_name);
for i = 1:imax - 1
  nedge = nedge + 1;
  edge(1,nedge) = i;
  edge(2,nedge) = i + 1;
  edge(3,nedge) = 0;
end

for i = 1:jmax - 1
  nedge = nedge + 1;
  edge(1,nedge) = i*imax;
  edge(2,nedge) = (i+1)*imax;
  edge(3,nedge) = 0;
end

for i = 1:imax - 1
  nedge = nedge + 1;
  edge(1,nedge) = imax*jmax - i + 1;
  edge(2,nedge) = imax*jmax - i;
  edge(3,nedge) = 0;
end

for i = 1:jmax - 1
  nedge = nedge + 1;
  edge(1,nedge) = (jmax-i)*imax + 1;
  edge(2,nedge) = (jmax-i-1)*imax + 1;
  edge(3,nedge) = 0;
end

numfix = zeros(0, 0, nag_int_name);
itrace = nag_int(1);
nqint = nag_int(10);

% Plot original mesh
fig1 = figure;
subplot(1,2,1);
triplot(transpose(double(conn(:,1:nelt))), coor(1,:), coor(2,:));
title ('Original Mesh');

[coor, ifail] = d06ca(coor, edge(:, 1:nedge), conn(:, 1:nelt), ...
                      numfix, itrace, nqint);

fprintf('\nComplete smooth mesh characteristics:\n');
fprintf('  nv    = %d\n', nv);
fprintf('  nelt  = %d\n', nelt);

% Plot smoothed mesh
subplot(1,2,2);
triplot(transpose(double(conn(:,1:nelt))), coor(1,:), coor(2,:));
title ('Smoothed Mesh');
```

9.2 Program Results

```
     d06ca example results

 BEFORE SMOOTHING
 Minimum smoothness measure:       1.0060557
 Maximum smoothness measure:      45.7310387
 Distribution interval            Number of elements
      1.0060557 -       5.4785540       715
```

```
      5.4785540 -       9.9510523         4
      9.9510523 -      14.4235506         1
     14.4235506 -      18.8960489         0
     18.8960489 -      23.3685472         0
     23.3685472 -      27.8410455         0
     27.8410455 -      32.3135438         0
     32.3135438 -      36.7860421         0
     36.7860421 -      41.2585404         0
     41.2585404 -      45.7310387         1

 AFTER SMOOTHING
 Minimum smoothness measure:       1.3377832
 Maximum smoothness measure:       1.4445226
 Distribution interval           Number of elements
      1.3377832 -       1.3484572         0
      1.3484572 -       1.3591311        13
      1.3591311 -       1.3698050        42
      1.3698050 -       1.3804790       104
      1.3804790 -       1.3911529       162
      1.3911529 -       1.4018268       159
      1.4018268 -       1.4125008       122
      1.4125008 -       1.4231747        74
      1.4231747 -       1.4338486        31
      1.4338486 -       1.4445226        14

Complete smooth mesh characteristics:
  nv    = 400
  nelt  = 722
```