

NAG Library Function Document

nag_mesh2d_delaunay (d06abc)

1 Purpose

nag_mesh2d_delaunay (d06abc) generates a triangular mesh of a closed polygonal region in \mathbb{R}^2 , given a mesh of its boundary. It uses a Delaunay–Voronoi process, based on an incremental method.

2 Specification

```
#include <nag.h>
#include <nagd06.h>

void nag_mesh2d_delaunay (Integer nvb, Integer nvint, Integer nvmax,
    Integer nedge, const Integer edge[], Integer *nv, Integer *nelt,
    double coor[], Integer conn[], const double weight[], Integer npropa,
    Integer itrace, const char *outfile, NagError *fail)
```

3 Description

nag_mesh2d_delaunay (d06abc) generates the set of interior vertices using a Delaunay–Voronoi process, based on an incremental method. It allows you to specify a number of fixed interior mesh vertices together with weights which allow concentration of the mesh in their neighbourhood. For more details about the triangulation method, 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 Arguments

- | | | |
|----|--|--------------|
| 1: | nvb – Integer | <i>Input</i> |
| | <i>On entry:</i> the number of vertices in the input boundary mesh. | |
| | <i>Constraint:</i> nvb \geq 3. | |
| 2: | nvint – Integer | <i>Input</i> |
| | <i>On entry:</i> the number of fixed interior mesh vertices to which a weight will be applied. | |
| | <i>Constraint:</i> nvint \geq 0. | |
| 3: | nvmax – Integer | <i>Input</i> |
| | <i>On entry:</i> the maximum number of vertices in the mesh to be generated. | |
| | <i>Constraint:</i> nvmax \geq nvb + nvint . | |
| 4: | nedge – Integer | <i>Input</i> |
| | <i>On entry:</i> the number of boundary edges in the input mesh. | |
| | <i>Constraint:</i> nedge \geq 1. | |

- 5: **edge**[$3 \times \mathbf{nedge}$] – const Integer *Input*
Note: the (i, j) th element of the matrix is stored in **edge** $[(j - 1) \times 3 + i - 1]$.
On entry: the specification of the boundary edges. **edge** $[(j - 1) \times 3]$ and **edge** $[(j - 1) \times 3 + 1]$ contain the vertex numbers of the two end points of the j th boundary edge. **edge** $[(j - 1) \times 3 + 2]$ is a user-supplied tag for the j th boundary edge and is not used by nag_mesh2d_delaunay (d06abc). Note that the edge vertices are numbered from 1 to **nvb**.
Constraint: $1 \leq \mathbf{edge}[(j - 1) \times 3 + i - 1] \leq \mathbf{nvb}$ and **edge** $[(j - 1) \times 3] \neq \mathbf{edge}[(j - 1) \times 3 + 1]$, for $i = 1, 2$ and $j = 1, 2, \dots, \mathbf{nedge}$.
- 6: **nv** – Integer * *Output*
On exit: the total number of vertices in the output mesh (including both boundary and interior vertices). If **nvb** + **nvint** = **nvmax**, no interior vertices will be generated and **nv** = **nvmax**.
- 7: **nelt** – Integer * *Output*
On exit: the number of triangular elements in the mesh.
- 8: **coor**[$2 \times \mathbf{nvmax}$] – double *Input/Output*
Note: the (i, j) th element of the matrix is stored in **coor** $[(j - 1) \times 2 + i - 1]$.
On entry: **coor** $[(i - 1) \times 2]$ contains the x coordinate of the i th input boundary mesh vertex, for $i = 1, 2, \dots, \mathbf{nvb}$. **coor** $[(i - 1) \times 2]$ contains the x coordinate of the $(i - \mathbf{nvb})$ th fixed interior vertex, for $i = \mathbf{nvb} + 1, \dots, \mathbf{nvb} + \mathbf{nvint}$. For boundary and interior vertices, **coor** $[(i - 1) \times 2 + 1]$ contains the corresponding y coordinate, for $i = 1, 2, \dots, \mathbf{nvb} + \mathbf{nvint}$.
On exit: **coor** $[(i - 1) \times 2]$ will contain the x coordinate of the $(i - \mathbf{nvb} - \mathbf{nvint})$ th generated interior mesh vertex, for $i = \mathbf{nvb} + \mathbf{nvint} + 1, \dots, \mathbf{nv}$; while **coor** $[(i - 1) \times 2 + 1]$ will contain the corresponding y coordinate. The remaining elements are unchanged.
- 9: **conn**[$3 \times (2 \times \mathbf{nvmax} + 5)$] – Integer *Output*
Note: the (i, j) th element of the matrix is stored in **conn** $[(j - 1) \times 3 + i - 1]$.
On exit: the connectivity of the mesh between triangles and vertices. For each triangle j , **conn** $[(j - 1) \times 3 + i - 1]$ gives the indices of its three vertices (in anticlockwise order), for $i = 1, 2, 3$ and $j = 1, 2, \dots, \mathbf{nelt}$. Note that the mesh vertices are numbered from 1 to **nv**.
- 10: **weight**[dim] – const double *Input*
Note: the dimension, dim , of the array **weight** must be at least $\max(1, \mathbf{nvint})$.
On entry: the weight of fixed interior vertices. It is the diameter of triangles (length of the longer edge) created around each of the given interior vertices.
Constraint: if **nvint** > 0, **weight** $[i - 1] > 0.0$, for $i = 1, 2, \dots, \mathbf{nvint}$.
- 11: **npropa** – Integer *Input*
On entry: the propagation type and coefficient, the argument **npropa** is used when the internal points are created. They are distributed in a geometric manner if **npropa** is positive and in an arithmetic manner if it is negative. For more details see Section 9.
Constraint: **npropa** $\neq 0$.
- 12: **itrace** – Integer *Input*
On entry: the level of trace information required from nag_mesh2d_delaunay (d06abc).
itrace ≤ 0
 No output is generated.

itrace ≥ 1

Output from the meshing solver is printed. This output contains details of the vertices and triangles generated by the process.

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

13: **outfile** – const char *

Input

On entry: the name of a file to which diagnostic output will be directed. If **outfile** is **NULL** the diagnostic output will be directed to standard output.

14: **fail** – NagError *

Input/Output

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.

NE_BAD_PARAM

On entry, argument $\langle value \rangle$ had an illegal value.

NE_INT

On entry, **nedge** = $\langle value \rangle$.

Constraint: **nedge** ≥ 1 .

On entry, **npropa** = 0.

On entry, **nvb** = $\langle value \rangle$.

Constraint: **nvb** ≥ 3 .

On entry, **nvint** = $\langle value \rangle$.

Constraint: **nvint** ≥ 0 .

NE_INT_2

On entry, the endpoints of the edge J have the same index I : $J = \langle value \rangle$ and $I = \langle value \rangle$.

NE_INT_3

On entry, **nvb** = $\langle value \rangle$, **nvint** = $\langle value \rangle$ and **nvmax** = $\langle value \rangle$.

Constraint: **nvmax** \geq **nvb** + **nvint**.

NE_INT_4

On entry, **EDGE**(I, J) = $\langle value \rangle$, $I = \langle value \rangle$, $J = \langle value \rangle$ and **nvb** = $\langle value \rangle$.

Constraint: **EDGE**(I, J) ≥ 1 and **EDGE**(I, J) \leq **nvb**, where **EDGE**(I, J) denotes **edge**[($J - 1$) \times 3 + $I - 1$].

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

NE_MESH_ERROR

An error has occurred during the generation of the boundary mesh. It appears that **nvmax** is not large enough: **nvmax** = $\langle value \rangle$.

An error has occurred during the generation of the interior mesh. Check the inputs of the boundary.

NE_NOT_CLOSE_FILE

Cannot close file $\langle value \rangle$.

NE_NOT_WRITE_FILE

Cannot open file $\langle value \rangle$ for writing.

NE_REAL_ARRAY_INPUT

On entry, $\mathbf{weight}[I - 1] = \langle value \rangle$ and $I = \langle value \rangle$.
Constraint: $\mathbf{weight}[I - 1] > 0.0$.

7 Accuracy

Not applicable.

8 Parallelism and Performance

`nag_mesh2d_delaunay (d06abc)` is not threaded by NAG in any implementation.

`nag_mesh2d_delaunay (d06abc)` makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The position of the internal vertices is a function position of the vertices on the given boundary. A fine mesh on the boundary results in a fine mesh in the interior. To dilute the influence of the data on the interior of the domain, the value of **npropa** can be changed. The propagation coefficient is calculated as:

$\omega = 1 + \frac{a - 1.0}{20.0}$, where a is the absolute value of **npropa**. During the process vertices are generated on edges of the mesh \mathcal{T}_i to obtain the mesh \mathcal{T}_{i+1} in the general incremental method (consult the d06 Chapter Introduction or George and Borouchaki (1998)). This generation uses the coefficient ω , and it is geometric if **npropa** > 0 , and arithmetic otherwise. But increasing the value of a may lead to failure of the process, due to precision, especially in geometries with holes. So you are advised to manipulate the argument **npropa** with care.

You are advised to take care to set the boundary inputs properly, especially for a boundary with multiply connected components. The orientation of the interior boundaries should be in **clockwise** order and opposite to that of the exterior boundary. If the boundary has only one connected component, its orientation should be **anticlockwise**.

10 Example

In this example, a geometry with two holes (two wings inside an exterior circle) is meshed using a Delaunay–Voronoi method. The exterior circle is centred at the point (1.0, 0.0) with a radius 3, the first RAE wing begins at the origin and it is normalized, and the last wing is a result from the first one after a translation, a scale reduction and a rotation. To be able to carry out some realistic computation on that geometry, some interior points have been introduced to have a finer mesh in the wake of those airfoils.

The boundary mesh has 296 vertices and 296 edges (see Figure 1 top). Note that the particular mesh generated could be sensitive to the *machine precision* and therefore may differ from one implementation to another. The interior meshes for different values of **npropa** are given in Figure 1.

10.1 Program Text

```

/* nag_mesh2d_delaunay (d06abc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagd06.h>

#define EDGE(I, J) edge[3*((J) -1)+(I) -1]
#define CONN(I, J) conn[3*((J) -1)+(I) -1]
#define COOR(I, J) coor[2*((J) -1)+(I) -1]

int main(void)
{
  const Integer nvmax = 6000, nvint = 40;
  double
    dnvint;
  Integer
    exit_status, i, itrace, j, k, nedge, nelt,
    npropa, nv, nvb, reftk, il;
  NagError
    fail;
  char
    pmesh[2];
  double
    *coor = 0, *weight = 0;
  Integer
    *conn = 0, *edge = 0;

  INIT_FAIL(fail);

  exit_status = 0;

  printf("nag_mesh2d_delaunay (d06abc) Example Program Results\n\n");

  /* Skip heading in data file */
  scanf("%*[\n] ");

  /* Reading of the geometry */
  scanf("%ld%ld%*[\n] ", &nvb, &nedge);

  if (nvb > nvmax)
  {
    printf("Problem with the array dimensions\n");
    printf(" nvb nvmax %6ld%6ld\n", nvb, nvmax);
    printf(" Please increase the value of nvmax\n");
    exit_status = -1;
    goto END;
  }

  /* Allocate memory */

  if (!(coor = NAG_ALLOC(2*nvmax, double)) ||
      !(weight = NAG_ALLOC(nvint, double)) ||
      !(conn = NAG_ALLOC(3*(2*nvmax + 5), Integer)) ||
      !(edge = NAG_ALLOC(3*nedge, Integer)))
  {
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
  }

  /* Coordinates of the boundary mesh vertices and boundary edges */

  for (i = 1; i <= nvb; ++i)
  {
    scanf("%ld", &il);
    scanf("%lf", &COOR(1, il));
    scanf("%lf", &COOR(2, il));
    scanf("%*[\n] ");
  }
}

```

```

for (i = 1; i <= nedge; ++i)
{
    scanf("%ld", &i1);
    scanf("%ld", &EDGE(1, i1));
    scanf("%ld", &EDGE(2, i1));
    scanf("%ld", &EDGE(3, i1));
    scanf("%*[\n] ");
}
scanf(" ' %1s '%*[\n]", pmesh);

/* Initialise mesh control parameters */

itrace = 0;

/* Generation of interior vertices on the RAE airfoils wake */

dnvint = 2.5/(double)(nvint + 1);

for (i = 1; i <= nvint; ++i)
{
    i1 = nvb + i;
    COOR(1, i1) = (double) i*dnvint + 1.38;
    COOR(2, i1) = -0.27*COOR(1, i1) + 0.2;
    weight[i-1] = 0.01;
}

/* Loop on the propagation coef */

for (j = 0; j < 4; ++j)
{
    switch (j)
    {
        case 0:
            npropa = -5;
            break;
        case 1:
            npropa = -1;
            break;
        case 2:
            npropa = 1;
            break;
        default:
            npropa = 5;
    }

    /* Call to the 2D Delaunay-Voronoi mesh generator */

    /* nag_mesh2d_delaunay (d06abc).
    * Generates a two-dimensional mesh using a Delaunay-Voronoi
    * process
    */
    nag_mesh2d_delaunay(nvb, nvint, nvmax, nedge, edge, &nv, &nelt, coor,
                       conn, weight, npropa, itrace, 0, &fail);

    if (fail.code == NE_NOERROR)
    {
        if (pmesh[0] == 'N')
        {
            printf(" Mesh characteristics with npropa =%6ld\n",
                   npropa);
            printf(" nv   =%6ld\n", nv);
            printf(" nelt =%6ld\n", nelt);
        }
        else if (pmesh[0] == 'Y')
        {
            /* Output the mesh to view it using the NAG Graphics Library */

            printf(" %10ld %10ld\n", nv, nelt);

            for (i = 1; i <= nv; ++i)

```

```

        {
            printf(" %15.6e %15.6e \n", COOR(1, i),
                COOR(2, i));
        }

        reftk = 0;
        for (k = 1; k <= nelt; ++k)
        {
            printf(" %10ld %10ld %10ld"
                " %10ld\n", CONN(1, k), CONN(2, k),
                CONN(3, k), reftk);
        }
    }
else
    {
        printf("Problem with the printing option Y or N\n");
        exit_status = -1;
        goto END;
    }
}
else
    {
        printf("Error from nag_mesh2d_delaunay (d06abc).\n%s\n",
            fail.message);
        exit_status = 1;
        goto END;
    }
}

END:
NAG_FREE(coor);
NAG_FREE(weight);
NAG_FREE(conn);
NAG_FREE(edge);

return exit_status;
}

```

10.2 Program Data

Note 1: since the data file for this example is quite large only a section of it is reproduced in this document. The full data file is distributed with your implementation.

```

nag_mesh2d_delaunay (d06abc) Example Program Data
      296      296      :NVB NEDGE
  1  0.400000E+01  0.000000E+00
  2  0.396307E+01  0.469303E+00
  3  0.385317E+01  0.927051E+00
  4  0.367302E+01  0.136197E+01
  5  0.342705E+01  0.176336E+01
  6  0.312132E+01  0.212132E+01
  7  0.276336E+01  0.242705E+01
  8  0.236197E+01  0.267302E+01
  9  0.192705E+01  0.285317E+01
 10  0.146930E+01  0.296307E+01
 11  0.100000E+01  0.300000E+01
 12  0.530697E+00  0.296307E+01
 13  0.729490E-01  0.285317E+01
 14  -.361971E+00  0.267302E+01
 15  -.763356E+00  0.242705E+01
 16  -.112132E+01  0.212132E+01
 17  -.142705E+01  0.176336E+01
 18  -.167302E+01  0.136197E+01
 19  -.185317E+01  0.927051E+00
 20  -.196307E+01  0.469303E+00
 21  -.200000E+01  0.367394E-15
 22  -.196307E+01  -.469303E+00
 23  -.185317E+01  -.927051E+00
 24  -.167302E+01  -.136197E+01
 25  -.142705E+01  -.176336E+01

```

26	-.112132E+01	-.212132E+01
27	-.763356E+00	-.242705E+01
28	-.361971E+00	-.267302E+01
29	0.729490E-01	-.285317E+01
30	0.530697E+00	-.296307E+01
31	0.100000E+01	-.300000E+01
32	0.146930E+01	-.296307E+01
33	0.192705E+01	-.285317E+01
34	0.236197E+01	-.267302E+01
35	0.276336E+01	-.242705E+01
36	0.312132E+01	-.212132E+01
37	0.342705E+01	-.176336E+01
38	0.367302E+01	-.136197E+01
39	0.385317E+01	-.927051E+00
40	0.396307E+01	-.469303E+00
41	0.000000E+00	0.000000E+00
42	0.602000E-03	0.316500E-02
43	0.240800E-02	0.630600E-02
44	0.541200E-02	0.941600E-02
45	0.960700E-02	0.124800E-01
46	0.149840E-01	0.154890E-01
47	0.215300E-01	0.184410E-01
48	0.292280E-01	0.213480E-01
49	0.380600E-01	0.242190E-01
50	0.480050E-01	0.270620E-01
51	0.590390E-01	0.298740E-01
52	0.711360E-01	0.326440E-01
53	0.842650E-01	0.353600E-01
54	0.983960E-01	0.380110E-01
55	0.113495E+00	0.405850E-01
56	0.129524E+00	0.430710E-01
57	0.146447E+00	0.454570E-01
58	0.164221E+00	0.477290E-01
59	0.182803E+00	0.498740E-01
60	0.202150E+00	0.518850E-01
61	0.222215E+00	0.537530E-01
62	0.242949E+00	0.554700E-01
63	0.264302E+00	0.570260E-01
64	0.286222E+00	0.584140E-01
65	0.308658E+00	0.596290E-01
66	0.331555E+00	0.606600E-01
67	0.354858E+00	0.614970E-01
68	0.378510E+00	0.621330E-01
69	0.402455E+00	0.625620E-01
70	0.426635E+00	0.627790E-01
71	0.450991E+00	0.627740E-01
72	0.475466E+00	0.625300E-01
73	0.500000E+00	0.620290E-01
74	0.524534E+00	0.612540E-01
75	0.549009E+00	0.601940E-01
76	0.573365E+00	0.588450E-01
77	0.597545E+00	0.572180E-01
78	0.621490E+00	0.553440E-01
79	0.645142E+00	0.532580E-01
80	0.668445E+00	0.509930E-01
81	0.691342E+00	0.485750E-01
82	0.713778E+00	0.460290E-01
83	0.735698E+00	0.433770E-01
84	0.757051E+00	0.406410E-01
85	0.777785E+00	0.378470E-01
86	0.797850E+00	0.350170E-01
87	0.817197E+00	0.321760E-01
88	0.835779E+00	0.293470E-01
89	0.853553E+00	0.265540E-01
90	0.870476E+00	0.238170E-01
91	0.886505E+00	0.211530E-01
92	0.901604E+00	0.185800E-01
93	0.915735E+00	0.161130E-01
94	0.928864E+00	0.137690E-01
95	0.940961E+00	0.115620E-01
96	0.951995E+00	0.950800E-02

97	0.961940E+00	0.762200E-02
98	0.970772E+00	0.591500E-02
99	0.978470E+00	0.440100E-02
100	0.985016E+00	0.309200E-02
101	0.990393E+00	0.200100E-02
102	0.994588E+00	0.113700E-02
103	0.997592E+00	0.510000E-03
104	0.999398E+00	0.128000E-03
105	0.100000E+01	0.000000E+00
106	0.999398E+00	0.350000E-04
107	0.997592E+00	0.137000E-03
108	0.994588E+00	0.296000E-03
109	0.990393E+00	0.497000E-03
110	0.985016E+00	0.719000E-03
111	0.978470E+00	0.935000E-03
112	0.970772E+00	0.111200E-02
113	0.961940E+00	0.121200E-02
114	0.951995E+00	0.119700E-02
115	0.940961E+00	0.103300E-02
116	0.928864E+00	0.694000E-03
117	0.915735E+00	0.157000E-03
118	0.901604E+00	-.600000E-03
119	0.886505E+00	-.159200E-02
120	0.870476E+00	-.282900E-02
121	0.853553E+00	-.431400E-02
122	0.835779E+00	-.604800E-02
123	0.817197E+00	-.802700E-02
124	0.797850E+00	-.102440E-01
125	0.777785E+00	-.126900E-01
126	0.757051E+00	-.153570E-01
127	0.735698E+00	-.182320E-01
128	0.713778E+00	-.212890E-01
129	0.691342E+00	-.244950E-01
130	0.668445E+00	-.278140E-01
131	0.645142E+00	-.312070E-01
132	0.621490E+00	-.346310E-01
133	0.597545E+00	-.380430E-01
134	0.573365E+00	-.413970E-01
135	0.549009E+00	-.446420E-01
136	0.524534E+00	-.477190E-01
137	0.500000E+00	-.505630E-01
138	0.475466E+00	-.530990E-01
139	0.450991E+00	-.552570E-01
140	0.426635E+00	-.569790E-01
141	0.402455E+00	-.582240E-01
142	0.378510E+00	-.589740E-01
143	0.354858E+00	-.592360E-01
144	0.331555E+00	-.590460E-01
145	0.308658E+00	-.584590E-01
146	0.286222E+00	-.575470E-01
147	0.264302E+00	-.563760E-01
148	0.242949E+00	-.549940E-01
149	0.222215E+00	-.534270E-01
150	0.202150E+00	-.516940E-01
151	0.182803E+00	-.498050E-01
152	0.164221E+00	-.477730E-01
153	0.146447E+00	-.456100E-01
154	0.129524E+00	-.433260E-01
155	0.113495E+00	-.409290E-01
156	0.983960E-01	-.384310E-01
157	0.842650E-01	-.358430E-01
158	0.711360E-01	-.331700E-01
159	0.590390E-01	-.304160E-01
160	0.480050E-01	-.275860E-01
161	0.380600E-01	-.246850E-01
162	0.292280E-01	-.217220E-01
163	0.215300E-01	-.187070E-01
164	0.149840E-01	-.156490E-01
165	0.960700E-02	-.125590E-01
166	0.541200E-02	-.944300E-02
167	0.240800E-02	-.630800E-02

168	0.602000E-03	-.316000E-02
169	0.991481E+00	-.647048E-01
170	0.992042E+00	-.635442E-01
171	0.993065E+00	-.625176E-01
172	0.994547E+00	-.616270E-01
173	0.996485E+00	-.608774E-01
174	0.998874E+00	-.602715E-01
175	0.100171E+01	-.598087E-01
176	0.100498E+01	-.594824E-01
177	0.100869E+01	-.592875E-01
178	0.101283E+01	-.592187E-01
179	0.101739E+01	-.592745E-01
180	0.102235E+01	-.594566E-01
181	0.102770E+01	-.597665E-01
182	0.103343E+01	-.602051E-01
183	0.103953E+01	-.607738E-01
184	0.104598E+01	-.614727E-01
185	0.105277E+01	-.623028E-01
186	0.105987E+01	-.632651E-01
187	0.106727E+01	-.643601E-01
188	0.107496E+01	-.655860E-01
189	0.108290E+01	-.669416E-01
190	0.109109E+01	-.684247E-01
191	0.109950E+01	-.700342E-01
192	0.110812E+01	-.717672E-01
193	0.111691E+01	-.736205E-01
194	0.112586E+01	-.755926E-01
195	0.113495E+01	-.776817E-01
196	0.114416E+01	-.798847E-01
197	0.115346E+01	-.821979E-01
198	0.116282E+01	-.846173E-01
199	0.117223E+01	-.871408E-01
200	0.118166E+01	-.897689E-01
201	0.119109E+01	-.925024E-01
202	0.120049E+01	-.953418E-01
203	0.120983E+01	-.982852E-01
204	0.121910E+01	-.101328E+00
205	0.122828E+01	-.104460E+00
206	0.123734E+01	-.107663E+00
207	0.124626E+01	-.110917E+00
208	0.125503E+01	-.114205E+00
209	0.126362E+01	-.117510E+00
210	0.127203E+01	-.120816E+00
211	0.128022E+01	-.124110E+00
212	0.128819E+01	-.127378E+00
213	0.129591E+01	-.130604E+00
214	0.130337E+01	-.133775E+00
215	0.131055E+01	-.136875E+00
216	0.131744E+01	-.139892E+00
217	0.132402E+01	-.142811E+00
218	0.133027E+01	-.145621E+00
219	0.133619E+01	-.148310E+00
220	0.134176E+01	-.150867E+00
221	0.134696E+01	-.153283E+00
222	0.135179E+01	-.155548E+00
223	0.135624E+01	-.157653E+00
224	0.136029E+01	-.159589E+00
225	0.136394E+01	-.161347E+00
226	0.136717E+01	-.162921E+00
227	0.136999E+01	-.164303E+00
228	0.137238E+01	-.165486E+00
229	0.137435E+01	-.166465E+00
230	0.137588E+01	-.167233E+00
231	0.137697E+01	-.167786E+00
232	0.137763E+01	-.168121E+00
233	0.137785E+01	-.168232E+00
234	0.137762E+01	-.168157E+00
235	0.137694E+01	-.167930E+00
236	0.137579E+01	-.167558E+00
237	0.137419E+01	-.167046E+00
238	0.137214E+01	-.166403E+00

```

239 0.136963E+01  -.165642E+00
240 0.136667E+01  -.164777E+00
241 0.136327E+01  -.163824E+00
242 0.135943E+01  -.162800E+00
243 0.135515E+01  -.161721E+00
244 0.135044E+01  -.160600E+00
245 0.134531E+01  -.159448E+00
246 0.133977E+01  -.158277E+00
247 0.133384E+01  -.157098E+00
248 0.132751E+01  -.155916E+00
249 0.132082E+01  -.154738E+00
250 0.131378E+01  -.153568E+00
251 0.130639E+01  -.152409E+00
252 0.129869E+01  -.151262E+00
253 0.129068E+01  -.150130E+00
254 0.128239E+01  -.149014E+00
255 0.127385E+01  -.147914E+00
256 0.126506E+01  -.146826E+00
257 0.125606E+01  -.145742E+00
258 0.124687E+01  -.144654E+00
259 0.123751E+01  -.143552E+00
260 0.122802E+01  -.142427E+00
261 0.121842E+01  -.141266E+00
262 0.120873E+01  -.140058E+00
263 0.119898E+01  -.138791E+00
264 0.118921E+01  -.137446E+00
265 0.117943E+01  -.136005E+00
266 0.116969E+01  -.134445E+00
267 0.116001E+01  -.132744E+00
268 0.115042E+01  -.130888E+00
269 0.114095E+01  -.128866E+00
270 0.113162E+01  -.126677E+00
271 0.112246E+01  -.124329E+00
272 0.111347E+01  -.121843E+00
273 0.110469E+01  -.119246E+00
274 0.109611E+01  -.116571E+00
275 0.108776E+01  -.113849E+00
276 0.107966E+01  -.111105E+00
277 0.107181E+01  -.108353E+00
278 0.106423E+01  -.105606E+00
279 0.105695E+01  -.102873E+00
280 0.104999E+01  -.100164E+00
281 0.104334E+01  -.974884E-01
282 0.103704E+01  -.948540E-01
283 0.103110E+01  -.922684E-01
284 0.102552E+01  -.897401E-01
285 0.102033E+01  -.872772E-01
286 0.101553E+01  -.848852E-01
287 0.101114E+01  -.825688E-01
288 0.100717E+01  -.803330E-01
289 0.100363E+01  -.781826E-01
290 0.100053E+01  -.761234E-01
291 0.997863E+00  -.741615E-01
292 0.995651E+00  -.723023E-01
293 0.993893E+00  -.705518E-01
294 0.992595E+00  -.689135E-01
295 0.991759E+00  -.673913E-01
296 0.991387E+00  -.659880E-01  :(I1, COOR(:,I1),I=1,...,NVB)
1 1 2 0
2 2 3 0
3 3 4 0
4 4 5 0
5 5 6 0
6 6 7 0
7 7 8 0
8 8 9 0
9 9 10 0
10 10 11 0
11 11 12 0
12 12 13 0
13 13 14 0

```

14	14	15	0
15	15	16	0
16	16	17	0
17	17	18	0
18	18	19	0
19	19	20	0
20	20	21	0
21	21	22	0
22	22	23	0
23	23	24	0
24	24	25	0
25	25	26	0
26	26	27	0
27	27	28	0
28	28	29	0
29	29	30	0
30	30	31	0
31	31	32	0
32	32	33	0
33	33	34	0
34	34	35	0
35	35	36	0
36	36	37	0
37	37	38	0
38	38	39	0
39	39	40	0
40	40	1	0
41	41	42	0
42	42	43	0
43	43	44	0
44	44	45	0
45	45	46	0
46	46	47	0
47	47	48	0
48	48	49	0
49	49	50	0
50	50	51	0
51	51	52	0
52	52	53	0
53	53	54	0
54	54	55	0
55	55	56	0
56	56	57	0
57	57	58	0
58	58	59	0
59	59	60	0
60	60	61	0
61	61	62	0
62	62	63	0
63	63	64	0
64	64	65	0
65	65	66	0
66	66	67	0
67	67	68	0
68	68	69	0
69	69	70	0
70	70	71	0
71	71	72	0
72	72	73	0
73	73	74	0
74	74	75	0
75	75	76	0
76	76	77	0
77	77	78	0
78	78	79	0
79	79	80	0
80	80	81	0
81	81	82	0
82	82	83	0
83	83	84	0
84	84	85	0

85	85	86	0
86	86	87	0
87	87	88	0
88	88	89	0
89	89	90	0
90	90	91	0
91	91	92	0
92	92	93	0
93	93	94	0
94	94	95	0
95	95	96	0
96	96	97	0
97	97	98	0
98	98	99	0
99	99	100	0
100	100	101	0
101	101	102	0
102	102	103	0
103	103	104	0
104	104	105	0
105	105	106	0
106	106	107	0
107	107	108	0
108	108	109	0
109	109	110	0
110	110	111	0
111	111	112	0
112	112	113	0
113	113	114	0
114	114	115	0
115	115	116	0
116	116	117	0
117	117	118	0
118	118	119	0
119	119	120	0
120	120	121	0
121	121	122	0
122	122	123	0
123	123	124	0
124	124	125	0
125	125	126	0
126	126	127	0
127	127	128	0
128	128	129	0
129	129	130	0
130	130	131	0
131	131	132	0
132	132	133	0
133	133	134	0
134	134	135	0
135	135	136	0
136	136	137	0
137	137	138	0
138	138	139	0
139	139	140	0
140	140	141	0
141	141	142	0
142	142	143	0
143	143	144	0
144	144	145	0
145	145	146	0
146	146	147	0
147	147	148	0
148	148	149	0
149	149	150	0
150	150	151	0
151	151	152	0
152	152	153	0
153	153	154	0
154	154	155	0
155	155	156	0

156	156	157	0
157	157	158	0
158	158	159	0
159	159	160	0
160	160	161	0
161	161	162	0
162	162	163	0
163	163	164	0
164	164	165	0
165	165	166	0
166	166	167	0
167	167	168	0
168	168	41	0
169	169	170	0
170	170	171	0
171	171	172	0
172	172	173	0
173	173	174	0
174	174	175	0
175	175	176	0
176	176	177	0
177	177	178	0
178	178	179	0
179	179	180	0
180	180	181	0
181	181	182	0
182	182	183	0
183	183	184	0
184	184	185	0
185	185	186	0
186	186	187	0
187	187	188	0
188	188	189	0
189	189	190	0
190	190	191	0
191	191	192	0
192	192	193	0
193	193	194	0
194	194	195	0
195	195	196	0
196	196	197	0
197	197	198	0
198	198	199	0
199	199	200	0
200	200	201	0
201	201	202	0
202	202	203	0
203	203	204	0
204	204	205	0
205	205	206	0
206	206	207	0
207	207	208	0
208	208	209	0
209	209	210	0
210	210	211	0
211	211	212	0
212	212	213	0
213	213	214	0
214	214	215	0
215	215	216	0
216	216	217	0
217	217	218	0
218	218	219	0
219	219	220	0
220	220	221	0
221	221	222	0
222	222	223	0
223	223	224	0
224	224	225	0
225	225	226	0
226	226	227	0

```

227 227 228 0
228 228 229 0
229 229 230 0
230 230 231 0
231 231 232 0
232 232 233 0
233 233 234 0
234 234 235 0
235 235 236 0
236 236 237 0
237 237 238 0
238 238 239 0
239 239 240 0
240 240 241 0
241 241 242 0
242 242 243 0
243 243 244 0
244 244 245 0
245 245 246 0
246 246 247 0
247 247 248 0
248 248 249 0
249 249 250 0
250 250 251 0
251 251 252 0
252 252 253 0
253 253 254 0
254 254 255 0
255 255 256 0
256 256 257 0
257 257 258 0
258 258 259 0
259 259 260 0
260 260 261 0
261 261 262 0
262 262 263 0
263 263 264 0
264 264 265 0
265 265 266 0
266 266 267 0
267 267 268 0
268 268 269 0
269 269 270 0
270 270 271 0
271 271 272 0
272 272 273 0
273 273 274 0
274 274 275 0
275 275 276 0
276 276 277 0
277 277 278 0
278 278 279 0
279 279 280 0
280 280 281 0
281 281 282 0
282 282 283 0
283 283 284 0
284 284 285 0
285 285 286 0
286 286 287 0
287 287 288 0
288 288 289 0
289 289 290 0
290 290 291 0
291 291 292 0
292 292 293 0
293 293 294 0
294 294 295 0
295 295 296 0
296 296 169 0

```

:(I1, EDGE(:,I1), I=1,...,NEDGE)
'N' :Printing option 'Y' or 'N'

10.3 Program Results

nag_mesh2d_delaunay (d06abc) Example Program Results

```
Mesh characteristics with npropa =   -5
nv   = 2317
nelt = 4340
Mesh characteristics with npropa =   -1
nv   = 4420
nelt = 8546
Mesh characteristics with npropa =    1
nv   = 5081
nelt = 9868
Mesh characteristics with npropa =    5
nv   = 2015
nelt = 3736
```

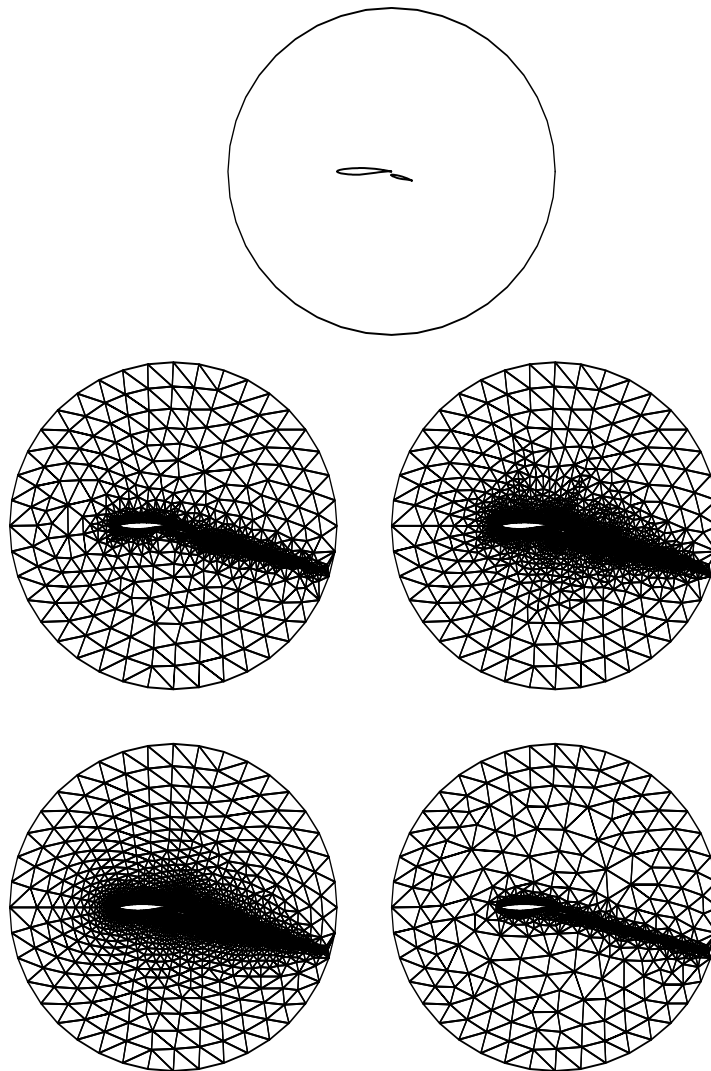


Figure 1

The boundary mesh (top), the interior mesh with **npropa** = -5 (middle left), -1 (middle right), 1 (bottom left) and 5 (bottom right) of a double RAE wings inside a circle geometry
