NAG Library Routine Document

D06ABF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

D06ABF 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

```
SUBROUTINE D06ABF (NVB, NVINT, NVMAX, NEDGE, EDGE, NV, NELT, COOR, CONN,        &
                   WEIGHT, NPROPA, ITRACE, RWORK, LRWORK, IWORK, LIWORK,        &
                   IFAIL)

INTEGER            NVB, NVINT, NVMAX, NEDGE, EDGE(3,NEDGE), NV, NELT,          &
                   CONN(3,2*NVMAX+5), NPROPA, ITRACE, LRWORK,                  &
                   IWORK(LIWORK), LIWORK, IFAIL
REAL (KIND=nag_wp) COOR(2,NVMAX), WEIGHT(*), RWORK(LRWORK)
```

3 Description

D06ABF 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 routine 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

1: NVB – INTEGER *Input*

On entry: the number of vertices in the input boundary mesh.

Constraint: NVB ≥ 3.

2: NVINT – INTEGER *Input*

On entry: the number of fixed interior mesh vertices to which a weight will be applied.

Constraint: NVINT ≥ 0.

3: NVMAX – INTEGER *Input*

On entry: the maximum number of vertices in the mesh to be generated.

Constraint: NVMAX $\geq$ NVB $+$ NVINT.

4: NEDGE – INTEGER *Input*

On entry: the number of boundary edges in the input mesh.

Constraint: NEDGE ≥ 1.

5: EDGE(3,NEDGE) – INTEGER array *Input*

On entry: the specification of the boundary 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 edge and is not used by D06ABF.

Constraint: $1 \leq \text{EDGE}(i,j) \leq \text{NVB}$ and $\text{EDGE}(1,j) \neq \text{EDGE}(2,j)$, for $i = 1,2$ and $j = 1,2,\ldots,\text{NEDGE}$.

6: NV – INTEGER *Output*

On exit: the total number of vertices in the output mesh (including both boundary and interior vertices). If $\text{NVB} + \text{NVINT} = \text{NVMAX}$, no interior vertices will be generated and $\text{NV} = \text{NVMAX}$.

7: NELT – INTEGER *Output*

On exit: the number of triangular elements in the mesh.

8: COOR(2,NVMAX) – REAL (KIND=nag_wp) array *Input/Output*

On entry: COOR$(1,i)$ contains the x coordinate of the ith input boundary mesh vertex, for $i = 1,2,\ldots,\text{NVB}$. COOR$(1,i)$ contains the x coordinate of the $(i - \text{NVB})$th fixed interior vertex, for $i = \text{NVB} + 1,\ldots,\text{NVB} + \text{NVINT}$. For boundary and interior vertices, COOR$(2,i)$ contains the corresponding y coordinate, for $i = 1,2,\ldots,\text{NVB} + \text{NVINT}$.

On exit: COOR$(1,i)$ will contain the x coordinate of the $(i - \text{NVB} - \text{NVINT})$th generated interior mesh vertex, for $i = \text{NVB} + \text{NVINT} + 1,\ldots,\text{NV}$; while COOR$(2,i)$ will contain the corresponding y coordinate. The remaining elements are unchanged.

9: CONN(3,$2 \times \text{NVMAX} + 5$) – INTEGER array *Output*

On exit: 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 $j = 1,2,\ldots,\text{NELT}$.

10: WEIGHT$(*)$ – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array WEIGHT must be at least $\max(1,\text{NVINT})$.

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 $\text{NVINT} > 0$, $\text{WEIGHT}(i) > 0.0$, for $i = 1,2,\ldots,\text{NVINT}$.

11: NPROPA – INTEGER *Input*

On entry: the propagation type and coefficient, the parameter 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 8.

Constraint: NPROPA $\neq 0$.

12: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D06ABF.

ITRACE ≤ 0
    No output is generated.

ITRACE ≥ 1

Output from the meshing solver is printed on the current advisory message unit (see X04ABF). 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: RWORK(LRWORK) – REAL (KIND=nag_wp) array *Workspace*

14: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which D06ABF is called.

Constraint: LRWORK $\geq 12 \times \text{NVMAX} + 15$.

15: IWORK(LIWORK) – INTEGER array *Workspace*

16: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which D06ABF is called.

Constraint: LIWORK $\geq 6 \times \text{NEDGE} + 32 \times \text{NVMAX} + 2 \times \text{NVB} + 78$.

17: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, NVB < 3,
or NVINT < 0,
or NVB + NVINT $>$ NVMAX,
or NEDGE < 1,
or EDGE$(i,j) < 1$ or EDGE$(i,j) >$ NVB, 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 NPROPA $= 0$;
or if NVINT > 0, WEIGHT$(i) \leq 0.0$, for some $i = 1,2,\ldots,$NVINT;
or LRWORK $< 12 \times \text{NVMAX} + 15$,
or LIWORK $< 6 \times \text{NEDGE} + 32 \times \text{NVMAX} + 2 \times \text{NVB} + 78$.

IFAIL $= 2$

An error has occurred during the generation of the interior mesh. Check the definition of the boundary (arguments COOR and EDGE) as well as the orientation of the boundary (especially in the case of a multiple connected component boundary). Setting ITRACE > 0 may provide more details.

IFAIL = 3

An error has occurred during the generation of the boundary mesh. It appears that NVMAX is not large enough.

7 Accuracy

Not applicable.

8 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 parameter 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**.

9 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.

9.1 Program Text

```
    Program d06abfe

!     D06ABF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: d06abf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: dnvint
      Integer                          :: i, i1, ifail, itrace, j, k, liwork,  &
                                          lrwork, nedge, nelt, npropa, nv,     &
                                          nvb, nvint, nvmax, reftk
      Character (1)                    :: pmesh
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: coor(:,:), rwork(:), weight(:)
      Integer, Allocatable             :: conn(:,:), edge(:,:), iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: real
```

```
!     .. Executable Statements ..
      Write (nout,*) 'D06ABF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

!     Reading of the geometry
!     Coordinates of the boundary mesh vertices and
!     edges references.

      Read (nin,*) nvb, nvint, nvmax, nedge
      lrwork = 12*nvmax + 15
      liwork = 6*nedge + 32*nvmax + 2*nvb + 78
      Allocate (coor(2,nvmax),rwork(lrwork),weight(nvint),conn(3,2*nvmax+5), &
        edge(3,nedge),iwork(liwork))

      Do i = 1, nvb
        Read (nin,*) i1, coor(1,i), coor(2,i)
      End Do

!     Boundary edges

      Do i = 1, nedge
        Read (nin,*) i1, edge(1,i), edge(2,i), edge(3,i)
      End Do

      Read (nin,*) pmesh

!     Initialise mesh control parameters

      itrace = 0

!     Generation of interior vertices on the
!     RAE airfoil's wake

      dnvint = 2.5E0_nag_wp/real(nvint+1,kind=nag_wp)

      Do i = 1, nvint
        i1 = nvb + i
        coor(1,i1) = 1.38E0_nag_wp + real(i,kind=nag_wp)*dnvint
        coor(2,i1) = -0.27E0_nag_wp*coor(1,i1) + 0.2E0_nag_wp
      End Do

      weight(1:nvint) = 0.01E0_nag_wp

      Write (nout,*)

!      Loop on the propagation coef

pcoef: Do j = 1, 4

        Select Case (j)
        Case (1)
          npropa = -5
        Case (2)
          npropa = -1
        Case (3)
          npropa = 1
        Case Default
          npropa = 5
        End Select

!       Call to the 2D Delaunay-Voronoi mesh generator

        ifail = 0
        Call d06abf(nvb,nvint,nvmax,nedge,edge,nv,nelt,coor,conn,weight, &
          npropa,itrace,rwork,lrwork,iwork,liwork,ifail)

        Select Case (pmesh)
        Case ('N')
          Write (nout,99999) 'Mesh characteristics with NPROPA =', npropa
```

```
          Write (nout,99999) 'NV   =', nv
          Write (nout,99999) 'NELT =', nelt
        Case ('Y')

!         Output the mesh

          Write (nout,99998) nv, nelt

          Do i = 1, nv
            Write (nout,99997) coor(1,i), coor(2,i)
          End Do

          reftk = 0

          Do k = 1, nelt
            Write (nout,99996) conn(1,k), conn(2,k), conn(3,k), reftk
          End Do

        Case Default
          Write (nout,*) 'Problem with the printing option Y or N'
          Exit pcoef
        End Select

      End Do pcoef

99999 Format (1X,A,I6)
99998 Format (1X,2I10)
99997 Format (2(2X,E13.6))
99996 Format (1X,4I10)
    End Program d06abfe
```

9.2 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.

```
D06ABF Example Program Data
       296       296     :NVB NEDGE
     1  0.400000E+01  0.000000E+00
            .
            .
            .
   296  0.991387E+00  -.659880E-01 :(I1, COOR(:,I),I=1,...,NVB)
   1   1   2   0
            .
            .
            .
 296 296 169   0  :(I1, EDGE(:,I), I=1,...,NEDGE)
'N'               :Printing option 'Y' or 'N'
```

9.3 Program Results

```
 D06ABF Example Program Results

 Mesh characteristics with NPROPA =    -5
 NV   =  2322
 NELT =  4350
 Mesh characteristics with NPROPA =    -1
 NV   =  4418
 NELT =  8542
 Mesh characteristics with NPROPA =     1
 NV   =  5071
 NELT =  9848
 Mesh characteristics with NPROPA =     5
 NV   =  1999
 NELT =  3704
```

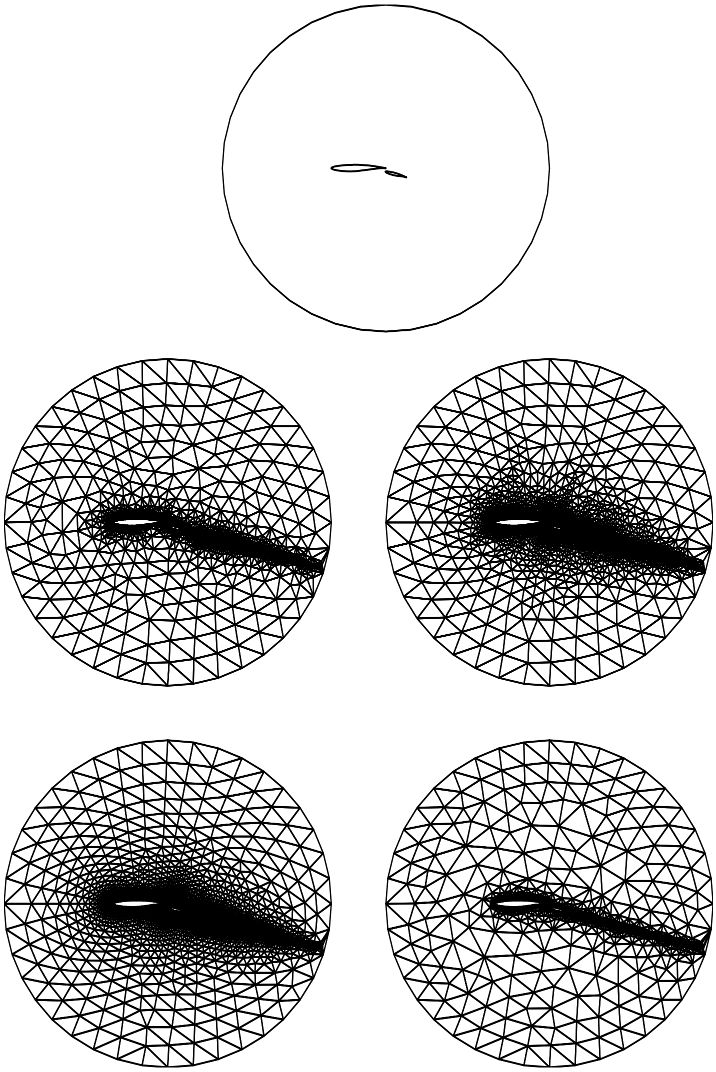

Figure 1
The boundary mesh (top), the interior mesh with NPROPA = −5 (middle left), −1 (middle right), 1 (bottom left) and 5 (bottom right) of a double RAE wings inside a circle geometry