NAG Fortran 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,
     1                  WEIGHT, NPROPA, ITRACE, RWORK, LRWORK, IWORK, LIWORK,
     2                  IFAIL)
      INTEGER           NVB, NVINT, NVMAX, NEDGE, EDGE(3,NEDGE), NV, NELT,
     1                  CONN(3,2*NVMAX+5), NPROPA, ITRACE, LRWORK,
     2                  IWORK(LIWORK), LIWORK, IFAIL
      real              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 the user 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 $\geq$ 3.

2: NVINT – INTEGER *Input*

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

Constraint: NVINT $\geq$ 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. $\mathrm{EDGE}(1:2, j)$ contains the vertex number of the two end-points of the jth boundary edge. $\mathrm{EDGE}(3, j)$ is a user-supplied tag for the jth boundary edge and is not used by this routine.

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

6: NV – INTEGER *Output*

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

7: NELT – INTEGER *Output*

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

8: COOR(2,NVMAX) – ***real*** array *Input/Output*

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

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

9: CONN(3,2*NVMAX+5) – INTEGER array *Output*

On exit: the connectivity of the mesh between triangles and vertices. For each triangle j, $\mathrm{CONN}(i, j)$ gives the indices in COOR of its three vertices (in anticlockwise order), for $i = 1, 2, 3$ and $j = 1, \ldots, \mathrm{NELT}$.

10: WEIGHT($*$) – ***real*** array *Input*

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

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

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 a arithmetic manner if it is negative. For more details see Section 8.

Constraint: $\mathrm{NPROPA} \neq 0$.

12: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D06ABF as follows:

if ITRACE ≤ 0, no output is generated;

if ITRACE ≥ 1, then 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.

Users are advised to set ITRACE $= 0$, unless they are experienced with the Finite Element meshes.

13: RWORK(LRWORK) – ***real*** 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 $\geq 12 \times$ 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 $\geq 6 \times$ NEDGE $+ 32 \times$ NVMAX $+ 2 \times$ NVB $+ 78$.

17: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

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, for users 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.**

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

7 Accuracy

Not applicable.

8 Further Comments

The position of the internal vertices is a function of the vertices positions on the given boundary. A fine mesh at the level of the boundary introduces a fine mesh of 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: $\omega = 1 + \frac{a - 1.0}{20.0}$, 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 the process to fail, due to precision problem, especially in some geometry with holes inside. So the user is advised to manipulate the argument NPROPA with care.

To ensure correct functioning of the routine, the user is advised 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 normalised, 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 kind of 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

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     D06ABF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NBEDMX, NVMAX, NVIMX, LRWORK, LIWORK
      PARAMETER        (NBEDMX=300,NVMAX=6000,NVIMX=40,
     +                 LRWORK=12*NVMAX+15,LIWORK=6*NBEDMX+34*NVMAX+78)
*     .. Local Scalars ..
      real             DNVINT
      INTEGER          I, I1, IFAIL, ITRACE, J, K, NEDGE, NELT, NPROPA,
     +                 NV, NVB, NVINT, REFTK
      CHARACTER        PMESH
*     .. Local Arrays ..
      real             COOR(2,NVMAX), RWORK(LRWORK), WEIGHT(NVIMX)
      INTEGER          CONN(3,2*NVMAX+5), EDGE(3,NBEDMX), IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         D06ABF
*     .. Intrinsic Functions ..
      INTRINSIC        real
```

```
*     .. Executable Statements ..
*
      WRITE (NOUT,*) 'D06ABF Example Program Results'
      WRITE (NOUT,*)
*
*     Skip heading in data file
*
      READ (NIN,*)
*
*     Reading of the geometry
*     Coordinates of the boundary mesh vertices and
*     edges references.
*
      READ (NIN,*) NVB, NEDGE
*
      IF (NVB.GT.NVMAX .OR. NEDGE.GT.NBEDMX) THEN
         WRITE (NOUT,*) 'Problem with the array dimensions '
         WRITE (NOUT,99999) ' NVB MAX ', NVB, NVMAX
         WRITE (NOUT,99999) ' NEDGE MAX ', NEDGE, NBEDMX
         STOP
      END IF
*
      DO 20 I = 1, NVB
         READ (NIN,*) I1, COOR(1,I), COOR(2,I)
   20 CONTINUE
*
*     Boundary edges
*
      DO 40 I = 1, NEDGE
         READ (NIN,*) I1, EDGE(1,I), EDGE(2,I), EDGE(3,I)
   40 CONTINUE
*
      READ (NIN,*) PMESH
*
*     Initialise mesh control parameters
*
      ITRACE = 0
*
*     Generation of interior vertices on the
*     RAE airfoils wake
*
      NVINT = 40
      DNVINT = 2.5e0/real(NVINT+1)
      DO 60 I = 1, NVINT
         I1 = NVB + I
         COOR(1,I1) = 1.38e0 + real(I)*DNVINT
         COOR(2,I1) = -0.27e0*COOR(1,I1) + 0.2e0
         WEIGHT(I) = 0.01e0
   60 CONTINUE
*
*     Loop on the propagation coef
*
      DO 120 J = 1, 4
         IF (J.EQ.1) THEN
            NPROPA = -5
         ELSE IF (J.EQ.2) THEN
            NPROPA = -1
         ELSE IF (J.EQ.3) THEN
            NPROPA = 1
         ELSE
            NPROPA = 5
         END IF
*
*     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)
*
```

```
         IF (PMESH.EQ.'N') THEN
            WRITE (NOUT,99998) 'Mesh characteristics with NPROPA =',
     +        NPROPA
            WRITE (NOUT,99998) 'NV   =', NV
            WRITE (NOUT,99998) 'NELT =', NELT
         ELSE IF (PMESH.EQ.'Y') THEN
*
*     Output the mesh to view it using the NAG Graphics Library
*
            WRITE (NOUT,99997) NV, NELT
            DO 80 I = 1, NV
               WRITE (NOUT,99996) COOR(1,I), COOR(2,I)
   80       CONTINUE
*
            REFTK = 0
            DO 100 K = 1, NELT
               WRITE (NOUT,99995) CONN(1,K), CONN(2,K), CONN(3,K), REFTK
  100       CONTINUE
         ELSE
            WRITE (NOUT,*) 'Problem with the printing option Y or N'
            STOP
         END IF
  120 CONTINUE
*
      STOP
*
99999 FORMAT (1X,A,2I6)
99998 FORMAT (1X,A,I6)
99997 FORMAT (1X,2I10)
99996 FORMAT (2(2X,E12.6))
99995 FORMAT (1X,4I10)
      END
```

9.2 Program Data

Note: 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   =  2319
 NELT =  4344
 Mesh characteristics with NPROPA =    -1
 NV   =  4421
 NELT =  8548
 Mesh characteristics with NPROPA =     1
 NV   =  5081
 NELT =  9868
 Mesh characteristics with NPROPA =     5
 NV   =  2015
 NELT =  3736
```

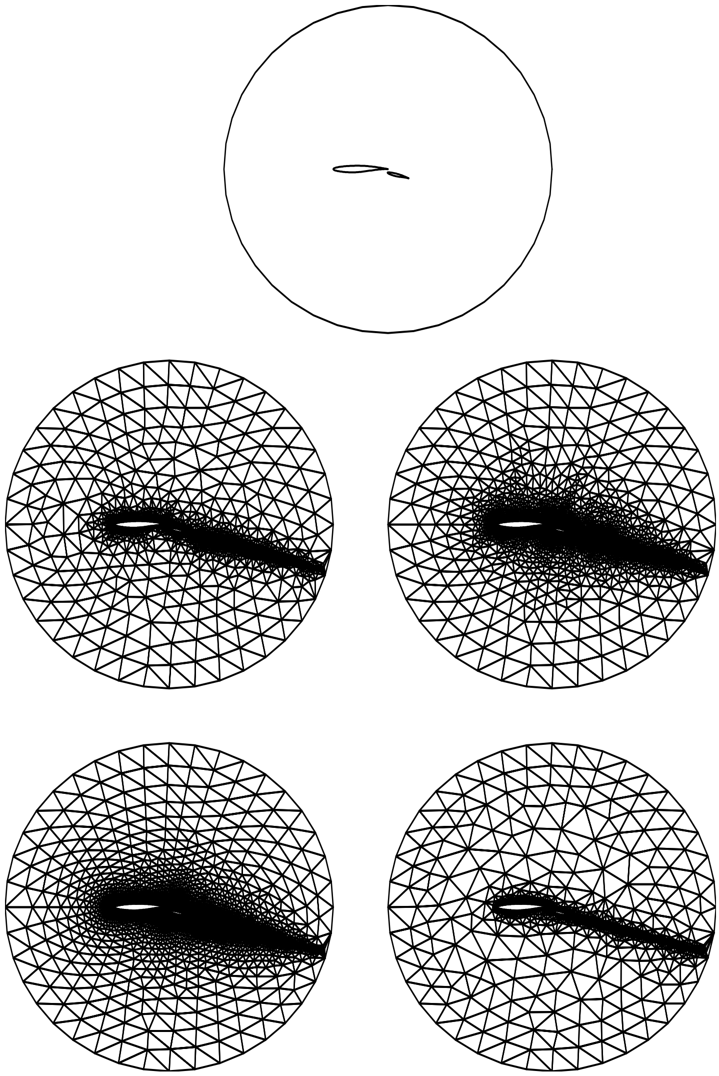

Figure 1
The boundary mesh (top), the interior mesh with NPROPA = −5 (middle left), −1 (middle right), 1 (bottom left) and 5 (bottom right) of a double RAE wings inside a circle geometry

---