NAG Fortran Library Routine Document

D06CAF

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

D06CAF uses a barycentering technique to smooth a given mesh.

2 Specification

```
 SUBROUTINE D06CAF(NV, NELT, NEDGE, COOR, EDGE, CONN, NVFIX, NUMFIX,
1                  ITRACE, NQINT, IWORK, LIWORK, RWORK, LRWORK, IFAIL)
 INTEGER           NV, NELT, NEDGE, EDGE(3,NEDGE), CONN(3,NELT), NVFIX,
1                  NUMFIX(*), ITRACE, NQINT, IWORK(LIWORK), LIWORK,
2                  LRWORK, IFAIL
 real              COOR(2,NV), RWORK(LRWORK)
```

3 Description

D06CAF 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 longer edge) of K, ρ_K is the radius of its inscribed circle and $\alpha = \frac{\sqrt{3}}{6}$ is a normalisation 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.

D06CAF makes small perturbation to vertices (using a barycenter formula) in order to give a reasonable good value of Q_K for all neighboring triangles. Some vertices may optionally be excluded from this process.

For more details about the triangulation smoothing method, especially with regards of different quality, 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: NV – INTEGER *Input*

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

Constraint: NV ≥ 3.

2: NELT – INTEGER *Input*

On entry: the number of triangles in the input mesh.

Constraint: NELT $\leq 2 \times$ NV $- 1$.

3: NEDGE – INTEGER *Input*

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

Constraint: NEDGE ≥ 1.

4: COOR(2,NV) – ***real*** array *Input/Output*

On entry: COOR$(1, i)$ contains the x-coordinate of the ith input mesh vertex, for $i = 1, \ldots, \mathrm{NV}$; while COOR$(2, i)$ contains the corresponding y-coordinate.

On exit: COOR$(1, i)$ will contain the x-coordinate of the ith smoothed mesh vertex, for $i = 1, \ldots, \mathrm{NV}$; 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 the user (see the description of NUMFIX), are unchanged by the process.

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

On entry: the specification of the boundary or interface edges. EDGE$(1:2, j)$ contains the vertex number 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 non-zero tag otherwise.

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

6: CONN(3,NELT) – INTEGER array *Input*

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

Constraint: $1 \leq \mathrm{CONN}(i, j) \leq \mathrm{NV}$, $\mathrm{CONN}(1, j) \neq \mathrm{CONN}(2, j)$, $\mathrm{CONN}(1, j) \neq \mathrm{CONN}(3, j)$ and $\mathrm{CONN}(2, j) \neq \mathrm{CONN}(3, j)$, for $i = 1, 2, 3$; $j = 1, \ldots, \mathrm{NELT}$.

7: NVFIX – INTEGER *Input*

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

Constraint: $0 \leq \mathrm{NVFIX} \leq \mathrm{NV}$.

8: NUMFIX($*$) – INTEGER array *Input*

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

On entry: the indices in COOR of fixed interior vertices of the input mesh.

Constraint: if NVFIX > 0, then $1 \leq \mathrm{NUMFIX}(i) \leq \mathrm{NV}$, for $i = 1, \ldots, \mathrm{NVFIX}$.

9: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D06CAF as follows:

if ITRACE ≤ 0, no output is generated;

if ITRACE $= 1$, then a histogram of the triangular element qualities is printed on the current advisory message unit (see X04ABF) 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;

if ITRACE > 1, then 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.

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

10: NQINT – INTEGER *Input*

On entry: the number of intervals between the extreme quality values for the input and the smoothed mesh. If ITRACE $= 0$, then NQINT is not referenced.

11: IWORK(LIWORK) – INTEGER array *Workspace*
12: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which D06CAF is called.

Constraint: LIWORK $\geq 8 \times \text{NELT} + 2 \times \text{NV}$.

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 D06CAF is called.

Constraint: LRWORK $\geq 2 \times \text{NV} + \text{NELT}$.

15: 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, NV < 3,
or NELT $> 2 \times \text{NV} - 1$,
or NEDGE < 1,
or EDGE$(i,j) < 1$ or EDGE$(i,j) >$ NV for some $i = 1, 2$ and $j = 1, \ldots, \text{NEDGE}$,
or EDGE$(1,j) =$ EDGE$(2,j)$ for some $j = 1, \ldots, \text{NEDGE}$,
or CONN$(i,j) < 1$ or CONN$(i,j) >$ NV for some $i = 1, 2, 3$ and $j = 1, \ldots, \text{NELT}$,
or CONN$(1,j) =$ CONN$(2,j)$ or CONN$(1,j) =$ CONN$(3,j)$ or CONN$(2,j) =$ CONN$(3,j)$ for some $j = 1, \ldots, \text{NELT}$,
or NVFIX < 0 or NVFIX $>$ NV,
or NUMFIX$(i) < 1$ or NUMFIX$(i) >$ NV for some $i = 1, \ldots, \text{NVFIX}$ if NVFIX > 0,
or LIWORK $< 8 \times \text{NELT} + 2 \times \text{NV}$,
or LRWORK $< 2 \times \text{NV} + \text{NELT}$.

IFAIL $= 2$

A serious error has occurred in an internal call to an auxiliary routine. 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.

7 Accuracy

Not applicable.

8 Further Comments

Not applicable.

9 Example

In this example, a uniform mesh on the unit square is randomly distorted using routines from Chapter G05 (Figure 1). D06CAF is then used to smooth the distorted mesh and recover a uniform mesh (Figure 2).

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.

```
*     D06CAF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NBEDMX, NVMAX, NELTMAX, NVFIXMX, LNUME, LIWORK,
     +                 LRWORK
      PARAMETER        (NBEDMX=100,NVMAX=400,NELTMAX=2*NVMAX-1,
     +                 NVFIXMX=20,LNUME=3*NELTMAX,
     +                 LIWORK=2*NVMAX+5*NELTMAX+LNUME,
     +                 LRWORK=2*NVMAX+NELTMAX)
*     .. Local Scalars ..
      real             DELTA, HX, HY, PI, R, RAD, SK, THETA, X1, X2, X3,
     +                 Y1, Y2, Y3
      INTEGER          I, IFAIL, IMAX, IND, ITRACE, J, JMAX, K, ME1,
     +                 ME2, ME3, NEDGE, NELT, NQINT, NV, NVFIX, REFTK
      CHARACTER        PMESH
*     .. Local Arrays ..
      real             COOR(2,NVMAX), RWORK(LRWORK)
      INTEGER          CONN(3,NELTMAX), EDGE(3,NBEDMX), IWORK(LIWORK),
     +                 NUMFIX(NVFIXMX)
*     .. External Functions ..
      real             G05DAF
      EXTERNAL         G05DAF
*     .. External Subroutines ..
      EXTERNAL         D06CAF, G05CBF
*     .. Intrinsic Functions ..
      INTRINSIC        ATAN, COS, real, MIN, SIN
*     .. Executable Statements ..
*
      WRITE (NOUT,*) 'D06CAF Example Program Results'
      WRITE (NOUT,*)
*
*     Skip heading in data file
*
      READ (NIN,*)
*
*     Read IMAX and JMAX, the number of vertices
*     in the x and y directions respectively.
*
      READ (NIN,*) IMAX, JMAX
*
*     Read distortion percentage and calculate radius
*     of distortion neighbourhood so that cross-over
*     can only occur at 100% or greater.
*
      READ (NIN,*) DELTA
*
      NV = IMAX*JMAX
      IF (NV.GT.NVMAX) THEN
         WRITE (NOUT,99999) 'Dimension problem NV MAX ', NV, NVMAX
         STOP
      END IF
*
```

```
      READ (NIN,*) PMESH
*
      HX = 1.e0/real(IMAX-1)
      HY = 1.e0/real(JMAX-1)
      RAD = 0.01e0*DELTA*MIN(HX,HY)/2.e0
      PI = 4.e0*ATAN(1.e0)
      CALL G05CBF(0)
      IND = 0
*
*     Generate a simple uniform mesh and then distort it
*     randomly within the distortion neighbourhood of each
*     node.
*
      DO 40 J = 1, JMAX
         DO 20 I = 1, IMAX

            R = G05DAF(0.e0,RAD)
            THETA = G05DAF(0.e0,2*PI)
            IF (I.EQ.1 .OR. I.EQ.IMAX .OR. J.EQ.1 .OR. J.EQ.JMAX)
     +          R = 0.e0

            K = (J-1)*IMAX + I
            COOR(1,K) = real(I-1)*HX + R*COS(THETA)
            COOR(2,K) = real(J-1)*HY + R*SIN(THETA)

            IF (I.LT.IMAX .AND. J.LT.JMAX) THEN
               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 IF
   20    CONTINUE
   40 CONTINUE
*
      NELT = IND
*
      IF (PMESH.EQ.'N') THEN
         WRITE (NOUT,*) 'The complete distorted mesh characteristics'
         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 60 I = 1, NV
            WRITE (NOUT,99996) COOR(1,I), COOR(2,I)
   60    CONTINUE
      ELSE
         WRITE (NOUT,*) 'Problem with the printing option Y or N'
         STOP
      END IF
*
      REFTK = 0
      DO 80 K = 1, NELT
         ME1 = CONN(1,K)
         ME2 = CONN(2,K)
         ME3 = CONN(3,K)
*
         X1 = COOR(1,ME1)
         X2 = COOR(1,ME2)
         X3 = COOR(1,ME3)
         Y1 = COOR(2,ME1)
         Y2 = COOR(2,ME2)
         Y3 = COOR(2,ME3)
*
         SK = ((X2-X1)*(Y3-Y1)-(Y2-Y1)*(X3-X1))/2.e0
```

```
          IF (SK.LT.0.e0) THEN
             WRITE (NOUT,*)
     +         'Error the surface of the element is negative'
             WRITE (NOUT,99998) 'K = ', K
             WRITE (NOUT,99994) 'SK = ', SK
             STOP
          END IF
          IF (PMESH.EQ.'Y') WRITE (NOUT,99995) CONN(1,K), CONN(2,K),
     +       CONN(3,K), REFTK
   80 CONTINUE
*
*     Boundary edges
*
      NEDGE = 0
      DO 100 I = 1, IMAX - 1
         NEDGE = NEDGE + 1
         EDGE(1,NEDGE) = I
         EDGE(2,NEDGE) = I + 1
         EDGE(3,NEDGE) = 0
  100 CONTINUE
*
      DO 120 I = 1, JMAX - 1
         NEDGE = NEDGE + 1
         EDGE(1,NEDGE) = I*IMAX
         EDGE(2,NEDGE) = (I+1)*IMAX
         EDGE(3,NEDGE) = 0
  120 CONTINUE
*
      DO 140 I = 1, IMAX - 1
         NEDGE = NEDGE + 1
         EDGE(1,NEDGE) = IMAX*JMAX - I + 1
         EDGE(2,NEDGE) = IMAX*JMAX - I
         EDGE(3,NEDGE) = 0
  140 CONTINUE
*
      DO 160 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
  160 CONTINUE
*
      NVFIX = 0
      NUMFIX(1) = 0
      ITRACE = 1
      NQINT = 10
      IFAIL = 0
*
*     Call the smoothing routine
*
      CALL D06CAF(NV,NELT,NEDGE,COOR,EDGE,CONN,NVFIX,NUMFIX,ITRACE,
     +            NQINT,IWORK,LIWORK,RWORK,LRWORK,IFAIL)
*
      IF (PMESH.EQ.'N') THEN
         WRITE (NOUT,*) 'The complete smoothed mesh characteristics'
         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 180 I = 1, NV
            WRITE (NOUT,99996) COOR(1,I), COOR(2,I)
  180    CONTINUE
*
         REFTK = 0
         DO 200 K = 1, NELT
            WRITE (NOUT,99995) CONN(1,K), CONN(2,K), CONN(3,K), REFTK
  200    CONTINUE
      END IF
```

```
*
      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)
99994 FORMAT (1X,A,E12.6)
      END
```

9.2 Program Data

```
D06CAF Example Program Data
20 20    :IMAX JMAX
87.0     :DELTA
'N'      :Printing option 'Y' or 'N'
```

9.3 Program Results

```
 D06CAF Example Program Results

 The complete distorted mesh characteristics
 NV   =   400
 NELT =   722
 BEFORE SMOOTHING
 MINIMUM SMOOTHNESS MEASURE :       1.0048907
 MINIMUM SMOOTHNESS MEASURE :     133.2110681
 DISTRIBUTION
 INTERVAL                         NUMBER OF ELEMENTS
      1.0048907 -      14.2255084       720
     14.2255084 -      27.4461262         0
     27.4461262 -      40.6667439         0
     40.6667439 -      53.8873616         0
     53.8873616 -      67.1079794         0
     67.1079794 -      80.3285971         0
     80.3285971 -      93.5492149         0
     93.5492149 -     106.7698326         0
    106.7698326 -     119.9904504         0
    119.9904504 -     133.2110681         1

 AFTER SMOOTHING
 MINIMUM SMOOTHNESS MEASURE :       1.3346259
 MINIMUM SMOOTHNESS MEASURE :       1.4572261
 DISTRIBUTION
 INTERVAL                         NUMBER OF ELEMENTS
      1.3346259 -       1.3468859        10
      1.3468859 -       1.3591459        36
      1.3591459 -       1.3714060        46
      1.3714060 -       1.3836660       117
      1.3836660 -       1.3959260       186
      1.3959260 -       1.4081860       137
      1.4081860 -       1.4204460       106
      1.4204460 -       1.4327061        51
      1.4327061 -       1.4449661        28
      1.4449661 -       1.4572261         4
 The complete smoothed mesh characteristics
 NV   =   400
 NELT =   722
```

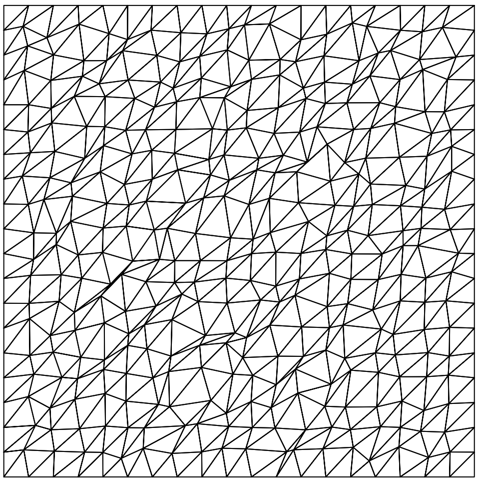

Figure 1
Distorted uniform mesh

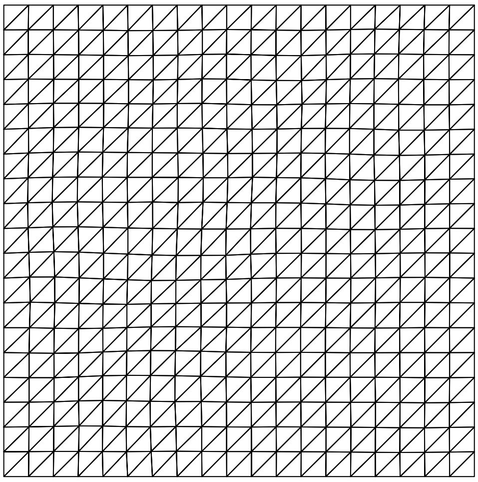

Figure 2
After smoothing with D06CAF