NAG 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
 double precision  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 longest edge) of K, ρ_K is the radius of its inscribed circle and $\alpha = \dfrac{\sqrt{3}}{6}$ 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.

D06CAF 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 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) – ***double precision*** array *Input/Output*

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

On exit: COOR$(1,i)$ will contain the x co-ordinate of the ith smoothed mesh vertex, for $i = 1, 2, \ldots, \text{NV}$; while COOR$(2,i)$ will contain the corresponding y co-ordinate. Note that the co-ordinates of boundary and interface edge vertices, as well as those specified by you (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$(1,j)$ 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: $1 \leq \text{EDGE}(i,j) \leq \text{NV}$ and $\text{EDGE}(1,j) \neq \text{EDGE}(2,j)$, for $i = 1, 2$ and $j = 1, 2, \ldots, \text{NEDGE}$.

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 of its three vertices (in anticlockwise order), for $i = 1, 2, 3$ and $j = 1, 2, \ldots, \text{NELT}$.

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

7: NVFIX – INTEGER *Input*

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

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

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

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

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

Constraint: if $\text{NVFIX} > 0$, $1 \leq \text{NUMFIX}(i) \leq \text{NV}$, for $i = 1, 2, \ldots, \text{NVFIX}$.

9: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D06CAF.

ITRACE ≤ 0
No output is generated.

ITRACE $= 1$
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.

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.

10: NQINT – INTEGER *Input*

On entry: the number of intervals between the extreme quality values for the input and the smoothed mesh.

If ITRACE $= 0$, 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 $\geq 8 \times$ NELT $+ 2 \times$ NV.

13: RWORK(LRWORK) – ***double precision*** 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 $\geq 2 \times$ NV $+$ NELT.

15: 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.

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, 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.**

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 $> 2 \times$ 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 routine. Check the input mesh, especially the connectivity between triangles and vertices (the parameter 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. D06CAF is then used to smooth the distorted mesh and recover a uniform mesh.

9.1 Program Text

```
*     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, MSTATE, LSEED
      PARAMETER        (NBEDMX=100,NVMAX=400,NELTMAX=2*NVMAX-1,
     +                 NVFIXMX=20,LNUME=3*NELTMAX,
     +                 LIWORK=2*NVMAX+5*NELTMAX+LNUME,
     +                 LRWORK=2*NVMAX+NELTMAX,MSTATE=200,LSEED=1)
*     .. Local Scalars ..
      DOUBLE PRECISION DELTA, HX, HY, PI2, R, RAD, SK, THETA, X1, X2,
     +                 X3, Y1, Y2, Y3
      INTEGER          GENID, I, IFAIL, IMAX, IND, ITRACE, J, JMAX, K,
     +                 LENSD, LSTATE, ME1, ME2, ME3, NEDGE, NELT, NQINT,
     +                 NV, NVFIX, REFTK, SUBID
      CHARACTER        PMESH
*     .. Local Arrays ..
      DOUBLE PRECISION COOR(2,NVMAX), RWORK(LRWORK)
      INTEGER          CONN(3,NELTMAX), EDGE(3,NBEDMX), IWORK(LIWORK),
     +                 NUMFIX(NVFIXMX), SEED(LSEED), STATE(MSTATE)
*     .. External Functions ..
      DOUBLE PRECISION X01AAF
      EXTERNAL         X01AAF
*     .. External Subroutines ..
      EXTERNAL         D06CAF, G05KFF, G05SQF
*     .. Intrinsic Functions ..
      INTRINSIC        COS, DBLE, MIN, SIN
*     .. Executable Statements ..
      CONTINUE
*
      LSTATE = MSTATE
      LENSD = LSEED
*
      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) NV, NVMAX
         GO TO 220
      END IF
*
      READ (NIN,*) PMESH
*
      HX = 1.0D0/DBLE(IMAX-1)
      HY = 1.0D0/DBLE(JMAX-1)
      RAD = 0.005D0*DELTA*MIN(HX,HY)
      PI2 = 2.0D0*X01AAF(PI2)
      IND = 0
*
*     Initialise the seed for the random number generator
      SEED(1) = 1762541
*     GENID and SUBID identify the base generator
      GENID = 1
      SUBID = 1
*     Initialise the generator to a repeatable sequence
      IFAIL = 1
      CALL G05KFF(GENID,SUBID,SEED,LENSD,STATE,LSTATE,IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,*)
         WRITE (NOUT,99993) ' ** G05KFF returned with IFAIL = ', IFAIL
         GO TO 220
      END IF
*     Generate two sets of uniform random variates
      IFAIL = 0
      CALL G05SQF(NV,0.0D0,RAD,STATE,RWORK,IFAIL)
      CALL G05SQF(NV,0.0D0,PI2,STATE,RWORK(NV+1),IFAIL)
*
*     Generate a simple uniform mesh and then distort it
*     randomly within the distortion neighbourhood of each
*     node.
*
      K = 0
      DO 40 J = 1, JMAX
         DO 20 I = 1, IMAX
*
            K = K + 1
            R = RWORK(K)
            THETA = RWORK(NV+K)
*
            IF (I.EQ.1 .OR. I.EQ.IMAX .OR. J.EQ.1 .OR. J.EQ.JMAX)
     +          R = 0.D0
*
            COOR(1,K) = DBLE(I-1)*HX + R*COS(THETA)
            COOR(2,K) = DBLE(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) 'Number of vertices =', NV
         WRITE (NOUT,99998) 'Number of elements =', 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: the printing option must be Y or N'
         GO TO 220
      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.D0
         IF (SK.LT.0.D0) THEN
            WRITE (NOUT,*)
     +        'Error: the surface of the element is negative'
            WRITE (NOUT,99998) 'element number  = ', K
            WRITE (NOUT,99994) 'element surface = ', SK
            GO TO 220
         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 = 1
*
*     Call the smoothing routine
*
      CALL D06CAF(NV,NELT,NEDGE,COOR,EDGE,CONN,NVFIX,NUMFIX,ITRACE,
     +            NQINT,IWORK,LIWORK,RWORK,LRWORK,IFAIL)
*
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,*)
         WRITE (NOUT,99993) ' ** D06CAF returned with IFAIL = ', IFAIL
         GO TO 220
      ELSE IF (PMESH.EQ.'N') THEN
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'The complete smoothed mesh characteristics'
         WRITE (NOUT,99998) 'Number of vertices =', NV
         WRITE (NOUT,99998) 'Number of elements =', 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
*
  220 CONTINUE
*
99999 FORMAT (1X,'Problem: the number of vertices, NV is ',I6,',',/'  ',
     +       '  the maximum allowed here is, NVMAX =',I6,'.')
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)
99993 FORMAT (1X,A,I5)
      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
 Number of vertices =   400
 Number of elements =   722
 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
```

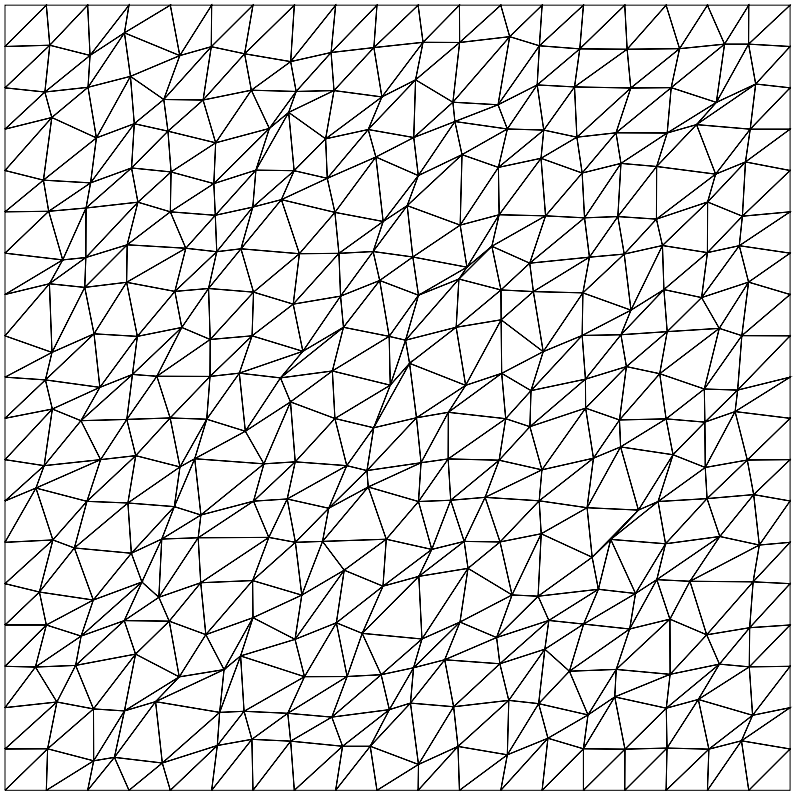
```
    36.7860421 -      41.2585404         0
    41.2585404 -      45.7310387         1

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

The complete smoothed mesh characteristics
Number of vertices =   400
Number of elements =   722
```

Example Program

Unform mesh on the unit square randomly distorted using routines from Chapter G05



Distorted mesh smoothed and a uniform mesh recovered