NAG Library Routine Document

D06CCF

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

D06CCF renumbers the vertices of a given mesh using a Gibbs method, in order the reduce the bandwidth of Finite Element matrices associated with that mesh.

2 Specification

```
SUBROUTINE D06CCF (NV, NELT, NEDGE, NNZMAX, NNZ, COOR, EDGE, CONN, IROW,        &
                   ICOL, ITRACE, IWORK, LIWORK, RWORK, LRWORK, IFAIL)

INTEGER            NV, NELT, NEDGE, NNZMAX, NNZ, EDGE(3,NEDGE),                 &
                   CONN(3,NELT), IROW(NNZMAX), ICOL(NNZMAX), ITRACE,            &
                   IWORK(LIWORK), LIWORK, LRWORK, IFAIL
REAL (KIND=nag_wp) COOR(2,NV), RWORK(LRWORK)
```

3 Description

D06CCF uses a Gibbs method to renumber the vertices of a given mesh in order to reduce the bandwidth of the associated finite element matrix A. This matrix has elements a_{ij} such that:

$$a_{ij} \neq 0 \Rightarrow i \text{ and } j \text{ are vertices belonging to the same triangle.}$$

This routine reduces the bandwidth m, which is the smallest integer such that $a_{ij} \neq 0$ whenever $|i-j| > m$ (see Gibbs *et al.* (1976) for details about that method). D06CCF also returns the sparsity structure of the matrix associated with the renumbered mesh.

This routine is derived from material in the MODULEF package from INRIA (Institut National de Recherche en Informatique et Automatique).

4 References

Gibbs N E, Poole W G Jr and Stockmeyer P K (1976) An algorithm for reducing the bandwidth and profile of a sparse matrix *SIAM J. Numer. Anal.* **13** 236–250

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 boundary edges in the input mesh.

Constraint: NEDGE ≥ 1.

4: NNZMAX – INTEGER *Input*

On entry: the maximum number of nonzero entries in the matrix based on the input mesh. It is the dimension of the arrays IROW and ICOL as declared in the subroutine from which D06CCF is called.

Constraint: $4 \times \text{NELT} + \text{NV} \leq \text{NNZMAX} \leq \text{NV}^2$.

5: NNZ – INTEGER *Output*

On exit: the number of nonzero entries in the matrix based on the input mesh.

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

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

On exit: $\text{COOR}(1,i)$ will contain the x coordinate of the ith renumbered mesh vertex, for $i = 1, 2, \ldots, \text{NV}$; while $\text{COOR}(2,i)$ will contain the corresponding y coordinate.

7: EDGE(3,NEDGE) – INTEGER array *Input/Output*

On entry: the specification of the boundary or interface edges. $\text{EDGE}(1,j)$ and $\text{EDGE}(2,j)$ contain the vertex numbers of the two end points of the jth boundary edge. $\text{EDGE}(3,j)$ is a user-supplied tag for the jth boundary or interface edge: $\text{EDGE}(3,j) = 0$ 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 $j = 1, 2, \ldots, \text{NEDGE}$.

On exit: the renumbered specification of the boundary or interface edges.

8: CONN(3,NELT) – INTEGER array *Input/Output*

On entry: the connectivity of the mesh between triangles and vertices. For each triangle j, $\text{CONN}(i,j)$ 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 $j = 1, 2, \ldots, \text{NELT}$.

On exit: the renumbered connectivity of the mesh between triangles and vertices.

9: IROW(NNZMAX) – INTEGER array *Output*
10: ICOL(NNZMAX) – INTEGER array *Output*

On exit: the first NNZ elements contain the row and column indices of the nonzero elements supplied in the finite element matrix A.

11: ITRACE – INTEGER *Input*

On entry: the level of trace information required from D06CCF.

ITRACE ≤ 0
No output is generated.

ITRACE $= 1$
Information about the effect of the renumbering on the finite element matrix are output. This information includes the half bandwidth and the sparsity structure of this matrix before and after renumbering.

ITRACE > 1
The output is similar to that produced when ITRACE $= 1$ but the sparsities (for each row of the matrix, indices of nonzero entries) of the matrix before and after renumbering are also output.

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

On entry: the dimension of the array IWORK as declared in the (sub)program from which D06CCF is called.

Constraint: LIWORK $\geq \max(\text{NNZMAX}, 20 \times \text{NV})$.

14: RWORK(LRWORK) – REAL (KIND=nag_wp) array *Workspace*
15: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which D06CCF is called.

Constraint: LRWORK $\geq$ NV.

16: 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, NV < 3,
or NELT $> 2 \times$ NV $- 1$,
or NEDGE < 1,
or NNZMAX $< 4 \times$ NELT $+$ NV or NNZMAX $>$ NV^2
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 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 LIWORK $< \max(\text{NNZMAX}, 20 \times \text{NV})$,
or LRWORK $<$ NV.

IFAIL $= 2$

A serious error has occurred during the computation of the compact sparsity of the finite element matrix or in an internal call to the renumbering routine. Check the input mesh, especially the connectivity between triangles and vertices (the parameter CONN). If the problem persists, contact NAG.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

In this example, a geometry with two holes (two interior circles inside an exterior one) is considered. The geometry has been meshed using the simple incremental method (D06AAF) and it has 250 vertices and 402 triangles (see Figure 1). The routine D06BAF is used to renumber the vertices, and one can see the benefit in terms of the sparsity of the finite element matrix based on the renumbered mesh (see Figure 2).

9.1 Program Text

```
    Program d06ccfe

!     D06CCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: d06cbf, d06ccf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, i1, ifail, itrace, k, liwork,     &
                                          lrwork, nedge, nelt, nnz, nnzmax,    &
                                          nv, reftk
      Character (1)                    :: pmesh
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: coor(:,:), rwork(:)
      Integer, Allocatable             :: conn(:,:), edge(:,:), icol(:),       &
                                          irow(:), iwork(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max
!     .. Executable Statements ..
      Write (nout,*) 'D06CCF Example Program Results'
      Flush (nout)

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

!     Reading of the geometry

      Read (nin,*) nv, nelt, nedge
      nnzmax = nv**2
      liwork = max(nnzmax,20*nv)
      lrwork = nv
      Allocate (conn(3,nelt),irow(nnzmax),icol(nnzmax),edge(3,nedge), &
        iwork(liwork),coor(2,nv),rwork(lrwork))

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

      Do k = 1, nelt
        Read (nin,*) conn(1,k), conn(2,k), conn(3,k), reftk
      End Do

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

!     Compute the sparsity of the FE matrix
!     from the input geometry

      ifail = 0
```

```
      Call d06cbf(nv,nelt,nnzmax,conn,nnz,irow,icol,ifail)

      Write (nout,*)

      Read (nin,*) pmesh

      Select Case (pmesh)
      Case ('N')
        Write (nout,*) 'The Matrix Sparsity characteristics'
        Write (nout,*) 'before the renumbering'
        Write (nout,99999) 'NV  =', nv
        Write (nout,99999) 'NNZ =', nnz
      Case ('Y')

!       Output the sparsity of the mesh

        Write (nout,99998) nv, nnz

        Do i = 1, nnz
          Write (nout,99998) irow(i), icol(i)
        End Do

      Case Default
        Write (nout,*) 'Problem with the printing option Y or N'
        Go To 100
      End Select

      Flush (nout)

!     Call the renumbering routine and get the new sparsity

      itrace = 1

      ifail = 0
      Call d06ccf(nv,nelt,nedge,nnzmax,nnz,coor,edge,conn,irow,icol,itrace, &
        iwork,liwork,rwork,lrwork,ifail)

      Select Case (pmesh)
      Case ('N')
        Write (nout,*)
        Write (nout,*) 'The Matrix Sparsity characteristics'
        Write (nout,*) 'after the renumbering'
        Write (nout,99999) 'NV   =', nv
        Write (nout,99999) 'NNZ  =', nnz
        Write (nout,99999) 'NELT =', nelt
      Case ('Y')

!       Output the sparsity of the renumbered mesh

        Write (nout,99998) nv, nnz

        Do i = 1, nnz
          Write (nout,99998) irow(i), icol(i)
        End Do

!       Output the renumbered 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

      End Select
```

```
100   Continue

99999 Format (1X,A,I6)
99998 Format (1X,2I10)
99997 Format (2(2X,E13.6))
99996 Format (1X,4I10)
    End Program d06ccfe
```

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.

```
D06CCF Example Program Data
       250       402       100       :NV NELT NEDGE
  0.100000E+01  0.000000E+00
            .
            .
            .
  0.112781E+00  0.103479E+00    :COOR(1:2,1:NV)
        21        55        56         1
            .
            .
            .
       151       250       155         1   :(CONN(:,K), REFT, K=1,...,NELT)
   1   1   2   1
            .
            .
            .
 100 100  71   1   :(I1, EDGE(:,I), I=1,NEDGE)
'N'                :Printing option 'Y' or 'N'
```

9.3 Program Results

```
 D06CCF Example Program Results

 The Matrix Sparsity characteristics
 before the renumbering
 NV  =   250
 NNZ =  1556

 INITIAL HALF-BAND-WIDTH:   234    INITIAL PROFILE: 18233
  FINAL HALF-BAND-WIDTH :    28     FINAL PROFILE  :  4038

 The Matrix Sparsity characteristics
 after the renumbering
 NV   =   250
 NNZ  =  1556
 NELT =   402
```

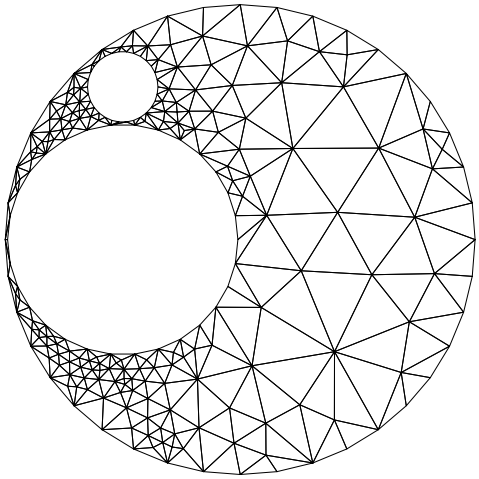

Figure 1
Mesh of the geometry

Figure 2
Sparsity of the matrix before (top) and after (bottom) the renumbering