

NAG Fortran Library Routine Document

D06DAF

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

D06DAF is a utility which performs an affine transformation of a given mesh.

2 Specification

```

SUBROUTINE D06DAF (NV, NEDGE, NELT, NTRANS, ITYPE, TRANS, COORI, EDGEI,
1 CONNI, COORO, EDGEO, CONNO, ITRACE, RWORK, LRWORK,
2 IFAIL)

INTEGER NV, NEDGE, NELT, NTRANS, ITYPE(NTRANS),
1 EDGEI(3,NEDGE), CONNI(3,NELT), EDGEO(3,NEDGE),
2 CONNO(3,NELT), ITRACE, LRWORK, IFAIL
double precision TRANS(6,NTRANS), COORI(2,NV), COORO(2,NV),
1 RWORK(LRWORK)

```

3 Description

D06DAF generates a mesh (co-ordinates, triangle/vertex connectivities and edge/vertex connectivities) resulting from an affine transformation of a given mesh. This transformation is of the form $Y = A \times X + B$, where

Y , X and B are in \mathbb{R}^2 , and

A is a real 2 by 2 matrix.

Such a transformation includes a translation, a rotation, a scale reduction or increase, a symmetric transformation with respect to a user-supplied line, a user-supplied analytic transformation, or a composition of several transformations.

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

4 References

None.

5 Parameters

- 1: NV – INTEGER *Input*
On entry: the total number of vertices in the input mesh.
Constraint: $NV \geq 3$.
- 2: NEDGE – INTEGER *Input*
On entry: the number of the boundary or interface edges in the input mesh.
Constraint: $NEDGE \geq 1$.
- 3: NELT – INTEGER *Input*
On entry: the number of triangles in the input mesh.
Constraint: $NELT \leq 2 \times NV - 1$.

- 4: NTRANS – INTEGER *Input*
On entry: the number of transformations of the input mesh.
Constraint: NTRANS \geq 1.
- 5: ITYPE(NTRANS) – INTEGER array *Input*
On entry: ITYPE(i), for $i = 1, \dots, \text{NTRANS}$, indicates the type of each transformation as follows:
 ITYPE(i) = 0
 Identity transformation.
 ITYPE(i) = 1
 Translation.
 ITYPE(i) = 2
 Symmetric transformation with respect to a user-supplied line.
 ITYPE(i) = 3
 Rotation.
 ITYPE(i) = 4
 Scaling.
 ITYPE(i) = 10
 User-supplied analytic transformation.
 Note that the transformations are applied in the order described in ITYPE.
Constraint: ITYPE(i) = 0, 1, 2, 3, 4 or 10, for $i = 1, 2, \dots, \text{NTRANS}$.
- 6: TRANS(6,NTRANS) – **double precision** array *Input*
On entry: the parameters for each transformation. For $i = 1, \dots, \text{NTRANS}$, TRANS(1, i) to TRANS(6, i) contain the parameters of the i th transformation.
 If ITYPE(i) = 0, elements TRANS(1, i) to TRANS(6, i) are not referenced.
 If ITYPE(i) = 1, the translation vector is $\vec{u} = \begin{pmatrix} a \\ b \end{pmatrix}$, where $a = \text{TRANS}(1, i)$ and $b = \text{TRANS}(2, i)$, while elements TRANS(3, i) to TRANS(6, i) are not referenced.
 If ITYPE(i) = 2, the user-supplied line is the curve $\{(x, y) \in \mathbb{R}^2; \text{ such that } ax + by + c = 0\}$, where $a = \text{TRANS}(1, i)$, $b = \text{TRANS}(2, i)$ and $c = \text{TRANS}(3, i)$, while elements TRANS(4, i) to TRANS(6, i) are not referenced.
 If ITYPE(i) = 3, the centre of the rotation is (x_0, y_0) where $x_0 = \text{TRANS}(1, i)$ and $y_0 = \text{TRANS}(2, i)$, $\theta = \text{TRANS}(3, i)$ is its angle in degrees, while elements TRANS(4, i) to TRANS(6, i) are not referenced.
 If ITYPE(i) = 4, $a = \text{TRANS}(1, i)$ is the scaling coefficient in the x -direction, $b = \text{TRANS}(2, i)$ is the scaling coefficient in the y -direction, and (x_0, y_0) are the scaling centre co-ordinates, with $x_0 = \text{TRANS}(3, i)$ and $y_0 = \text{TRANS}(4, i)$; while elements TRANS(5, i) to TRANS(6, i) are not referenced.
 If ITYPE(i) = 10, the user-supplied analytic affine transformation $Y = A \times X + B$ is such that $A = (a_{kl})_{1 \leq k, l \leq 2}$ and $B = (b_k)_{1 \leq k \leq 2}$ where $a_{kl} = \text{TRANS}(2 \times (k - 1) + l, i)$, and $b_k = \text{TRANS}(4 + k, i)$ with $k, l = 1, 2$.

- 7: COORI(2,NV) – *double precision* array *Input/Output*
On entry: COORI(1,*i*) contains the *x* co-ordinate of the *i*th vertex of the input mesh, for $i = 1, \dots, NV$; while COORI(2,*i*) contains the corresponding *y* co-ordinate.
On exit: see Section 8.
- 8: EDGEI(3,NEDGE) – INTEGER array *Input/Output*
On entry: the specification of the boundary or interface edges. EDGEI(1,*j*) and EDGEI(2,*j*) contain the vertex numbers of the two end points of the *j*th boundary edge. EDGEI(3,*j*) is a user-supplied tag for the *j*th boundary edge.
On exit: see Section 8.
Constraint: $1 \leq \text{EDGEI}(i, j) \leq NV$ and $\text{EDGEI}(1, j) \neq \text{EDGEI}(2, j)$, for $i = 1, 2$ and $j = 1, 2, \dots, NEDGE$.
- 9: CONNI(3,NELT) – INTEGER array *Input/Output*
On entry: the connectivity of the input mesh between triangles and vertices. For each triangle *j*, CONNI(*i*,*j*) gives the indices of its three vertices (in anticlockwise order), for $i = 1, 2, 3$ and $j = 1, \dots, NELT$.
On exit: see Section 8.
Constraints:
 $1 \leq \text{CONNI}(i, j) \leq NV$;
 $\text{CONNI}(1, j) \neq \text{CONNI}(2, j)$;
 $\text{CONNI}(1, j) \neq \text{CONNI}(3, j)$ and $\text{CONNI}(2, j) \neq \text{CONNI}(3, j)$, for $i = 1, 2, 3$ and $j = 1, 2, \dots, NELT$.
- 10: COORO(2,NV) – *double precision* array *Output*
On exit: COORO(1,*i*) will contain the *x* co-ordinate of the *i*th vertex of the transformed mesh, for $i = 1, \dots, NV$; while COORO(2,*i*) will contain the corresponding *y* co-ordinate.
- 11: EDGEO(3,NEDGE) – INTEGER array *Output*
On exit: the specification of the boundary or interface edges of the transformed mesh. If the number of symmetric transformations is even or zero then $\text{EDGEO}(i, j) = \text{EDGEI}(i, j)$, for $i = 1, 2, 3$ and $j = 1, \dots, NEDGE$; otherwise $\text{EDGEO}(1, j) = \text{EDGEI}(2, j)$, $\text{EDGEO}(2, j) = \text{EDGEI}(1, j)$ and $\text{EDGEO}(3, j) = \text{EDGEI}(3, j)$, for $j = 1, \dots, NEDGE$.
- 12: CONNO(3,NELT) – INTEGER array *Output*
On exit: the connectivity of the transformed mesh between triangles and vertices. If the number of symmetric transformations is even or zero then $\text{CONNO}(i, j) = \text{CONNI}(i, j)$, for $i = 1, 2, 3$ and $j = 1, \dots, NELT$; otherwise $\text{CONNO}(1, j) = \text{CONNI}(1, j)$, $\text{CONNO}(2, j) = \text{CONNI}(3, j)$ and $\text{CONNO}(3, j) = \text{CONNI}(2, j)$, for $j = 1, \dots, NELT$.
- 13: ITRACE – INTEGER *Input*
On entry: the level of trace information required from D06DAF.
 $\text{ITRACE} \leq 0$
 No output is generated.
 $\text{ITRACE} \geq 1$
 Details of each transformation, the matrix *A* and the vector *B* of the final transformation, which is the composition of all the NTRANS transformations, are printed on the current advisory message unit (see X04ABF).

- 14: RWORK(LRWORK) – *double precision* array *Workspace*
 15: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which D06DAF is called.

Constraint: LRWORK $\geq 12 \times$ NTRANS.

- 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 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, 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 EDGEI(*i,j*) < 1 or EDGEI(*i,j*) > NV for some *i* = 1, 2 and *j* = 1, ..., NEDGE;
 or EDGEI(1,*j*) = EDGEI(2,*j*) for some *j* = 1, ..., NEDGE;
 or CONNI(*i,j*) < 1 or CONNI(*i,j*) > NV for some *i* = 1, 2, 3 and *j* = 1, ..., NELT;
 or CONNI(1,*j*) = CONNI(2,*j*) or CONNI(1,*j*) = CONNI(3,*j*) or
 CONNI(2,*j*) = CONNI(3,*j*) for some *j* = 1, ..., NELT;
 or NTRANS < 1;
 or ITYPE(*i*) \neq 0, 1, 2, 3, 4 or 10 for some *i* = 1, ..., NTRANS;
 or LRWORK < 12 \times NTRANS.

IFAIL = 2

A serious error has occurred in an internal call to an auxiliary routine. Check the input mesh especially the triangles/vertices and the edges/vertices connectivities as well as the details of each transformations.

7 Accuracy

Not applicable.

8 Further Comments

You may not wish to save the input mesh (COORI, EDGEI and CONNI) and could call D06DAF using the same parameters for the input and the output (transformed) mesh.

9 Example

For an example of the use of this utility routine, see Section 9 of the document for D06DBF.