

# NAG Library Function Document

## nag\_mesh2d\_sparse (d06cbc)

### 1 Purpose

nag\_mesh2d\_sparse (d06cbc) generates the sparsity pattern of a finite element matrix associated with a given mesh.

### 2 Specification

```
#include <nag.h>
#include <nagd06.h>

void nag_mesh2d_sparse (Integer nv, Integer nelt, Integer nnzmax,
    const Integer conn[], Integer *nnz, Integer irow[], Integer icol[],
    NagError *fail)
```

### 3 Description

nag\_mesh2d\_sparse (d06cbc) generates the sparsity pattern of a finite element matrix associated with a given mesh. The sparsity pattern is returned in a coordinate storage format consistent with the sparse linear algebra functions in Chapter f11. More precisely nag\_mesh2d\_sparse (d06cbc) returns the number of nonzero elements in the associated sparse matrix, and their row and column indices. This is designed to assist you in applying finite element discretization to meshes from the d06 Chapter Introduction and in solving the resulting sparse linear system using functions from Chapter f11.

The output sparsity pattern is based on the fact that finite element matrix  $A$  has elements  $a_{ij}$  satisfying:

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

### 4 References

None.

### 5 Arguments

- |    |  |              |
|----|--|--------------|
| 1: | <b>nv</b> – Integer  | <i>Input</i> |
|    | <i>On entry:</i> the total number of vertices in the input mesh.   |              |
|    | <i>Constraint:</i> <b>nv</b> $\geq 3$ .  |              |
| 2: | <b>nelt</b> – Integer  | <i>Input</i> |
|    | <i>On entry:</i> the number of triangles in the input mesh.  |              |
|    | <i>Constraint:</i> <b>nelt</b> $\leq 2 \times \mathbf{nv} - 1$ .   |              |
| 3: | <b>nnzmax</b> – Integer  | <i>Input</i> |
|    | <i>On entry:</i> the maximum number of nonzero entries in the matrix based on the input mesh. It is the dimension of the arrays <b>irow</b> and <b>icol</b> as declared in the function from which nag_mesh2d_sparse (d06cbc) is called. |              |
|    | <i>Constraint:</i> $4 \times \mathbf{nelt} + \mathbf{nv} \leq \mathbf{nnzmax} \leq \mathbf{nv}^2$ .  |              |
| 4: | <b>conn</b> [ $3 \times \mathbf{nelt}$ ] – const Integer   | <i>Input</i> |
|    | <b>Note:</b> the $(i, j)$ th element of the matrix is stored in <b>conn</b> [( $j - 1$ ) $\times 3 + i - 1$ ].   |              |

*On entry:* the connectivity of the mesh between triangles and vertices. For each triangle  $j$ ,  $\mathbf{conn}[(j-1) \times 3 + i - 1]$  gives the indices of its three vertices (in anticlockwise order), for  $i = 1, 2, 3$  and  $j = 1, 2, \dots, \mathbf{nelt}$ . Note that the mesh vertices are numbered from 1 to  $\mathbf{nv}$ .

*Constraint:*  $1 \leq \mathbf{conn}[(j-1) \times 3 + i - 1] \leq \mathbf{nv}$  and  $\mathbf{conn}[(j-1) \times 3] \neq \mathbf{conn}[(j-1) \times 3 + 1]$  and  $\mathbf{conn}[(j-1) \times 3] \neq \mathbf{conn}[(j-1) \times 3 + 2]$  and  $\mathbf{conn}[(j-1) \times 3 + 1] \neq \mathbf{conn}[(j-1) \times 3 + 2]$ , for  $i = 1, 2, 3$  and  $j = 1, 2, \dots, \mathbf{nelt}$ .

5: **nnz** – Integer \* *Output*

*On exit:* the number of nonzero entries in the matrix associated with the input mesh.

6: **irow**[**nnzmax**] – Integer *Output*

7: **icol**[**nnzmax**] – Integer *Output*

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

8: **fail** – NagError \* *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.

See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

### NE\_BAD\_PARAM

On entry, argument  $\langle value \rangle$  had an illegal value.

### NE\_INT

On entry,  $\mathbf{nv} = \langle value \rangle$ .

Constraint:  $\mathbf{nv} \geq 3$ .

### NE\_INT\_2

On entry,  $\mathbf{nelt} = \langle value \rangle$  and  $\mathbf{nv} = \langle value \rangle$ .

Constraint:  $\mathbf{nelt} \leq 2 \times \mathbf{nv} - 1$ .

On entry,  $\mathbf{nnzmax} = \langle value \rangle$  and  $\mathbf{nv} = \langle value \rangle$ .

Constraint:  $\mathbf{nnzmax} \leq \mathbf{nv}^2$ .

On entry, vertices 1 and 2 of the triangle  $K$  have the same index  $I$ :  $K = \langle value \rangle$  and  $I = \langle value \rangle$ .

On entry, vertices 1 and 3 of the triangle  $K$  have the same index  $I$ :  $K = \langle value \rangle$  and  $I = \langle value \rangle$ .

On entry, vertices 2 and 3 of the triangle  $K$  have the same index  $I$ :  $K = \langle value \rangle$  and  $I = \langle value \rangle$ .

### NE\_INT\_3

On entry,  $\mathbf{nnzmax} = \langle value \rangle$ ,  $\mathbf{nelt} = \langle value \rangle$  and  $\mathbf{nv} = \langle value \rangle$ .

Constraint:  $\mathbf{nnzmax} \geq (4 \times \mathbf{nelt} + \mathbf{nv})$ .

### NE\_INT\_4

On entry,  $\mathbf{CONN}(I, J) = \langle value \rangle$ ,  $I = \langle value \rangle$ ,  $J = \langle value \rangle$  and  $\mathbf{nv} = \langle value \rangle$ .

Constraint:  $\mathbf{CONN}(I, J) \geq 1$  and  $\mathbf{CONN}(I, J) \leq \mathbf{nv}$ , where  $\mathbf{CONN}(I, J)$  denotes  $\mathbf{conn}[(J-1) \times 3 + I - 1]$ .

**NE\_INTERNAL\_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.  
See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

A serious error has occurred in an internal call to an auxiliary function. Check the input mesh especially the connectivity. Seek expert help.

**NE\_NO\_LICENCE**

Your licence key may have expired or may not have been installed correctly.  
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

**7 Accuracy**

Not applicable.

**8 Parallelism and Performance**

nag\_mesh2d\_sparse (d06cbc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the x06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

**9 Further Comments**

None.

**10 Example**

See Section 10 in nag\_mesh2d\_renum (d06ccc).

---