

NAG C 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

```
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 non-zero elements in the associated sparse matrix, and their row and column indices. This is designed to assist the user in applying finite element discretisation 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:

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

4 References

None.

5 Parameters

- 1: **nv** – Integer *Input*
On entry: the total number of vertices in the input mesh.
Constraint: $\mathbf{nv} \geq 3$.
- 2: **nelt** – Integer *Input*
On entry: the number of triangles in the input mesh.
Constraint: $\mathbf{nelt} \leq 2 \times \mathbf{nv} - 1$.
- 3: **nnzmax** – Integer *Input*
On entry: the maximum number of non-zero entries in the matrix based on the input mesh. It is the dimension of the arrays **irow** and **icol** as declared in the function from which nag_mesh2d_sparse (d06cbc) is called.
Constraint: $4 \times \mathbf{nelt} + \mathbf{nv} \leq \mathbf{nnzmax} \leq \mathbf{nv}^2$.
- 4: **conn**[$3 \times \mathbf{nelt}$] – const Integer *Input*
Note: where **CONN**(i, j) appears in this document it refers to the array element **conn**[$3 \times (j - 1) + i - 1$]. We recommend using a #define to make the same definition in your calling program.

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 $j = 1, \dots, \mathbf{nelt}$. Note that the mesh vertices are numbered from 1 to **nv**.

Constraint: $1 \leq \mathbf{CONN}(i, j) \leq \mathbf{nv}$ and $\mathbf{CONN}(1, j) \neq \mathbf{CONN}(2, j)$ and $\mathbf{CONN}(1, j) \neq \mathbf{CONN}(3, j)$ and $\mathbf{CONN}(2, j) \neq \mathbf{CONN}(3, j)$ for $i = 1, 2, 3$ and $j = 1, 2, \dots, \mathbf{nelt}$.

5: **nnz** – Integer * *Output*

On exit: the number of non-zero 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 non-zero elements supplied in the Finite Element matrix A .

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

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT

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

Constraint: **nv** ≥ 3 .

NE_INT_2

On entry, **nelt** = $\langle \text{value} \rangle$, **nv** = $\langle \text{value} \rangle$.

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

On entry, vertices 2 and 3 of the triangle k have the same index i : $k = \langle \text{value} \rangle$, $i = \langle \text{value} \rangle$.

On entry, vertices 1 and 3 of the triangle k have the same index i : $k = \langle \text{value} \rangle$, $i = \langle \text{value} \rangle$.

On entry, vertices 1 and 2 of the triangle k have the same index i : $k = \langle \text{value} \rangle$, $i = \langle \text{value} \rangle$.

On entry, **nnzmax** $> \mathbf{nv}^2$: **nnzmax** = $\langle \text{value} \rangle$, **nv** = $\langle \text{value} \rangle$.

NE_INT_3

On entry, **nnzmax** $< (4 \times \mathbf{nelt} + \mathbf{nv})$: **nnzmax** = $\langle \text{value} \rangle$, **nelt** = $\langle \text{value} \rangle$, **nv** = $\langle \text{value} \rangle$.

NE_INT_4

On entry, **conn**(i, j) < 1 or **conn**(i, j) $> \mathbf{nv}$, where **conn**(i, j) denotes **conn**[$3 \times (j - 1) + i - 1$]: **conn**(i, j) = $\langle \text{value} \rangle$, $i = \langle \text{value} \rangle$, $j = \langle \text{value} \rangle$, **nv** = $\langle \text{value} \rangle$.

NE_INTERNAL_ERROR

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

NE_BAD_PARAM

On entry, parameter $\langle \text{value} \rangle$ had an illegal value.

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 consult NAG for assistance.

7 Accuracy

Not applicable.

8 Further Comments

Not applicable.

9 Example

See Section 9 of the document for nag_mesh2d_renum (d06ccc).
