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: **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: **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 function from which nag_mesh2d_sparse (d06cbc) is called.

Constraint: $4 \times$ **nelt** $+$ **nv** $\leq$ **nnzmax** $\leq$ **nv**^{2}.

4: **conn**[**3** $\times$ **nelt**] – const Integer *Input*

Note: the (i, j)th element of the matrix is stored in **conn**$[(j-1) \times 3 + i - 1]$.

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

Constraint: $1 \leq$ **conn**$[(j-1)\times 3+i-1] \leq$ **nv** and **conn**$[(j-1)\times 3] \neq$ **conn**$[(j-1)\times 3+1]$ and **conn**$[(j-1)\times 3] \neq$ **conn**$[(j-1)\times 3+2]$ and **conn**$[(j-1)\times 3+1] \neq$ **conn**$[(j-1)\times 3+2]$, for $i=1,2,3$ and $j=1,2,\ldots,$ **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 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **nv** $=$ ⟨*value*⟩.
Constraint: **nv** ≥ 3.

NE_INT_2

On entry, **nelt** $=$ ⟨*value*⟩ and **nv** $=$ ⟨*value*⟩.
Constraint: **nelt** $\leq 2\times$ **nv** -1.

On entry, **nnzmax** $=$ ⟨*value*⟩ and **nv** $=$ ⟨*value*⟩.
Constraint: **nnzmax** $\leq$ **nv**^{2}.

On entry, vertices 1 and 2 of the triangle K have the same index I: $K=$ ⟨*value*⟩ and $I=$ ⟨*value*⟩.

On entry, vertices 1 and 3 of the triangle K have the same index I: $K=$ ⟨*value*⟩ and $I=$ ⟨*value*⟩.

On entry, vertices 2 and 3 of the triangle K have the same index I: $K=$ ⟨*value*⟩ and $I=$ ⟨*value*⟩.

NE_INT_3

On entry, **nnzmax** $=$ ⟨*value*⟩, **nelt** $=$ ⟨*value*⟩ and **nv** $=$ ⟨*value*⟩.
Constraint: **nnzmax** $\geq (4\times$ **nelt** $+$ **nv**$)$.

NE_INT_4

On entry, **CONN**$(I,J)=$ ⟨*value*⟩, $I=$ ⟨*value*⟩, $J=$ ⟨*value*⟩ and **nv** $=$ ⟨*value*⟩.
Constraint: **CONN**$(I,J)\geq 1$ and **CONN**$(I,J)\leq$ **nv**, where **CONN**(I,J) denotes **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.

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

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 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).