NAG Toolbox for MATLAB

f11zn

1 Purpose

f11zn sorts the nonzero elements of a complex sparse non-Hermitian matrix, represented in co-ordinate storage format.

2 Syntax

```
[nnz, a, irow, icol, istr, ifail] = f11zn(n, nnz, a, irow, icol, dup, zero)
```

3 Description

f11zn takes a co-ordinate storage (CS) representation (see Section 2.1.1 in the F11 Chapter Introduction) of a complex n by n sparse non-Hermitian matrix A, and reorders the nonzero elements by increasing row index and increasing column index within each row. Entries with duplicate row and column indices may be removed, or the values may be summed. Any entries with zero values may optionally be removed.

The function also returns a pointer array **istr** to the starting address of each row in A.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **n – int32 scalar**

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 1$.

2: **nnz – int32 scalar**

The number of nonzero elements in the matrix A.

Constraint: $\mathbf{nnz} \geq 0$.

3: **a**$(*)$ **– complex array**

Note: the dimension of the array **a** must be at least $\max(1, \mathbf{nnz})$.

The nonzero elements of the matrix A. These may be in any order and there may be multiple nonzero elements with the same row and column indices.

4: **irow**$(*)$ **– int32 array**

Note: the dimension of the array **irow** must be at least $\max(1, \mathbf{nnz})$.

The row indices corresponding to the nonzero elements supplied in the array **a**.

Constraint: $1 \leq \mathbf{irow}(i) \leq \mathbf{n}$, for $i = 1, 2, \ldots, \mathbf{nnz}$.

5: **icol**$(*)$ **– int32 array**

Note: the dimension of the array **icol** must be at least $\max(1, \mathbf{nnz})$.

The column indices corresponding to the nonzero elements supplied in the array **a**.

Constraint: $1 \leq \mathbf{icol}(i) \leq \mathbf{n}$, for $i = 1, 2, \ldots, \mathbf{nnz}$.

6: **dup – string**

Indicates how any nonzero elements with duplicate row and column indices are to be treated.

dup = 'R'

The entries are removed.

dup = 'S'

The relevant values in **a** are summed.

dup = 'F'

The function fails with **ifail** $= 3$ on detecting a duplicate.

Constraint: **dup** = 'R', 'S' or 'F'.

7: **zero – string**

Indicates how any elements with zero values in **a** are to be treated.

zero = 'R'

The entries are removed.

zero = 'K'

The entries are kept.

zero = 'F'

The function fails with **ifail** $= 4$ on detecting a zero.

Constraint: **zero** = 'R', 'K' or 'F'.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

iwork

5.4 Output Parameters

1: **nnz – int32 scalar**

The number of nonzero elements with unique row and column indices.

2: **a**$(*)$ **– complex array**

Note: the dimension of the array **a** must be at least $\max(1, \mathbf{nnz})$.

The nonzero elements ordered by increasing row index, and by increasing column index within each row. Each nonzero element has a unique row and column index.

3: **irow**$(*)$ **– int32 array**

Note: the dimension of the array **irow** must be at least $\max(1, \mathbf{nnz})$.

The first **nnz** elements contain the row indices corresponding to the nonzero elements returned in the array **a**.

4: **icol**$(*)$ **– int32 array**

Note: the dimension of the array **icol** must be at least $\max(1, \mathbf{nnz})$.

The first **nnz** elements contain the row indices corresponding to the nonzero elements returned in the array **a**.

5: **istr**$(\mathbf{n}+1)$ **– int32 array**

istr(i), for $i = 1, 2, \ldots, \mathbf{n}$, contains the index of arrays **a**, **irow** and **icol** where row i of the matrix A starts. **istr**$(\mathbf{n}+1)$ contains the index $+1$ of the last nonzero element in A. See also Section 8.

6: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **n** < 1,
or **nnz** < 0,
or **dup** $\neq$ 'R', 'S' or 'F',
or **zero** $\neq$ 'R', 'K' or 'F'.

ifail $= 2$

On entry, a nonzero element has been supplied which does not lie within the matrix A, i.e., one or more of the following constraints has been violated:

$1 \leq \mathbf{irow}(i) \leq \mathbf{n}$,

$1 \leq \mathbf{icol}(i) \leq \mathbf{n}$,

for $i = 1, 2, \ldots, \mathbf{nnz}$.

ifail $= 3$

On entry, **dup** $=$ 'F', and nonzero elements have been supplied which have duplicate row and column indices.

ifail $= 4$

On entry, **zero** $=$ 'F', and at least one matrix element has been supplied with a zero coefficient value.

7 Accuracy

Not applicable.

8 Further Comments

The time taken for a call to f11zn is proportional to **nnz**.

Note that the resulting matrix may have either rows or columns with no entries. If row i has no entries then **istr**$(i) =$ **istr**$(i+1)$.

9 Example

```
n = int32(5);
nnz = int32(15);
a = [complex(4, +1);
     complex(-2, +6);
     complex(1, -3);
     complex(-2, -1);
     complex(-3, +0);
     complex(1, +2);
     complex(0, +0);
     complex(1, +3);
     complex(-1, -1);
     complex(6, -3);
     complex(2, +6);
     complex(2, +1);
     complex(1, +0);
     complex(0, -3);
     complex(2, +2)];
irow = [int32(3);
     int32(5);
     int32(4);
     int32(4);
     int32(5);
     int32(1);
     int32(1);
     int32(3);
     int32(2);
     int32(5);
     int32(1);
     int32(4);
     int32(2);
     int32(3);
     int32(4)];
icol = [int32(1);
     int32(2);
     int32(4);
     int32(2);
     int32(5);
     int32(2);
     int32(5);
     int32(5);
     int32(4);
     int32(5);
     int32(1);
     int32(2);
     int32(3);
     int32(3);
     int32(5)];
dup = 'S';
zero = 'R';
[nnzOut, aOut, irowOut, icolOut, istr, ifail] = ...
    f11zn(n, nnz, a, irow, icol, dup, zero)
```

```
nnzOut =
          11
aOut =
   2.0000 + 6.0000i
   1.0000 + 2.0000i
   1.0000
  -1.0000 - 1.0000i
   4.0000 + 1.0000i
        0 - 3.0000i
   1.0000 + 3.0000i
   1.0000 - 3.0000i
   2.0000 + 2.0000i
  -2.0000 + 6.0000i
   3.0000 - 3.0000i
```

```
irowOut =
           1
           1
           2
           2
           3
           3
           3
           4
           4
           5
           5
icolOut =
           1
           2
           3
           4
           1
           3
           5
           4
           5
           2
           5
istr =
           1
           3
           5
           8
          10
          12
ifail =
           0
```