NAG Toolbox for MATLAB

f11xn

1 Purpose

f11xn computes a matrix-vector or conjugate transposed matrix-vector product involving a complex sparse non-Hermitian matrix stored in co-ordinate storage format.

2 Syntax

```
[y, ifail] = f11xn(trans, a, irow, icol, check, x, 'n', n, 'nnz', nnz)
```

3 Description

f11xn computes either the matrix-vector product $y = Ax$, or the conjugate transposed matrix-vector product $y = A^{\mathrm{H}}x$, according to the value of the argument **trans**, where A is a complex n by n sparse non-Hermitian matrix, of arbitrary sparsity pattern. The matrix A is stored in co-ordinate storage (CS) format (see Section 2.1.1 in the F11 Chapter Introduction). The array **a** stores all the nonzero elements of A, while arrays **irow** and **icol** store the corresponding row and column indices respectively.

It is envisaged that a common use of f11xn will be to compute the matrix-vector product required in the application of f11bs to sparse complex linear systems. This is illustrated in Section 9 of the document for f11dr.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **trans – string**

Specifies whether or not the matrix A is conjugate transposed.

trans = 'N'

$y = Ax$ is computed.

trans = 'T'

$y = A^{\mathrm{H}}x$ is computed.

Constraint: **trans** = 'N' or 'T'.

2: **a**(**nnz**) **– complex array**

The nonzero elements in the matrix A, ordered by increasing row index, and by increasing column index within each row. Multiple entries for the same row and column indices are not permitted. The function f11zn may be used to order the elements in this way.

3: **irow**(**nnz**) **– int32 array**

4: **icol**(**nnz**) **– int32 array**

The row and column indices of the nonzero elements supplied in **a**.

Constraints:

$1 \leq \mathbf{irow}(i) \leq \mathbf{n}$ and $1 \leq \mathbf{icol}(i) \leq \mathbf{n}$, for $i = 1, 2, \ldots, \mathbf{nnz}$;
$\mathbf{irow}(i-1) < \mathbf{irow}(i)$ or $\mathbf{irow}(i-1) = \mathbf{irow}(i)$ and $\mathbf{icol}(i-1) < \mathbf{icol}(i)$, for $i = 2, 3, \ldots, \mathbf{nnz}$.

5: **check – string**

Specifies whether or not the CS representation of the matrix A should be checked.

check = 'C'

Checks are carried on the values of **n**, **nnz**, **irow** and **icol**.

check = 'N'

None of these checks are carried out.

See also Section 8.2.

Constraint: **check** = 'C' or 'N'.

6: **x(n) – complex array**

The vector *x*.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the arrays **x**, **y**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix *A*.

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

2: **nnz – int32 scalar**

Default: The dimension of the arrays **a**, **irow**, **icol**. (An error is raised if these dimensions are not equal.)

the number of nonzero elements in the matrix *A*.

Constraint: $1 \leq \mathbf{nnz} \leq \mathbf{n}^2$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **y(n) – complex array**

The vector *y*.

2: **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, **trans** $\neq$ 'N' or 'T',
or **check** $\neq$ 'C' or 'N'.

ifail $= 2$

On entry, **n** < 1,
or **nnz** < 1,
or **nnz** $>$ **n**^{2}.

ifail $= 3$

On entry, the arrays **irow** and **icol** fail to satisfy the following constraints:

$1 \leq$ **irow**$(i) \leq$ **n** and $1 \leq$ **icol**$(i) \leq$ **n**, for $i = 1, 2, \ldots,$ **nnz**;

irow$(i-1) <$ **irow**(i) or **irow**$(i-1) =$ **irow**(i) and **icol**$(i-1) <$ **icol**(i), for $i = 2, 3, \ldots,$ **nnz**.

Therefore a nonzero element has been supplied which does not lie within the matrix A, is out of order, or has duplicate row and column indices. Call f11zn to reorder and sum or remove duplicates.

7 Accuracy

The computed vector y satisfies the error bound:

$\|y - Ax\|_\infty \leq c(n)\epsilon\|A\|_\infty\|x\|_\infty$, if **trans** = 'N', or

$\left\|y - A^{\mathrm{T}}x\right\|_\infty \leq c(n)\epsilon\left\|A^{\mathrm{T}}\right\|_\infty\|x\|_\infty$, if **trans** = 'T',

where $c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

8 Further Comments

8.1 Timing

The time taken for a call to f11xn is proportional to **nnz**.

8.2 Use of check

It is expected that a common use of f11xn will be to compute the matrix-vector product required in the application of f11bs to sparse complex linear systems. In this situation f11xn is likely to be called many times with the same matrix A. In the interests of both reliability and efficiency you are recommended to set **check** to 'C' for the first of such calls, and to 'N' for all subsequent calls.

9 Example

```
trans = 'N';
a = [complex(2, +3);
     complex(1, -4);
     complex(1, +0);
     complex(-1, -2);
     complex(4, +1);
     complex(0, +1);
     complex(1, +3);
     complex(0, -1);
     complex(2, -6);
     complex(-2, +0);
     complex(3, +1)];
irow = [int32(1);
     int32(1);
     int32(2);
     int32(2);
     int32(3);
     int32(3);
     int32(3);
     int32(4);
     int32(4);
     int32(5);
     int32(5)];
icol = [int32(1);
     int32(2);
     int32(3);
     int32(4);
     int32(1);
     int32(3);
     int32(5);
     int32(4);
     int32(5);
     int32(2);
     int32(5)];
check = 'C';
x = [complex(0.7, +0.21);
     complex(0.16, -0.43);
     complex(0.52, +0.97);
     complex(0.77, +0);
     complex(0.28, -0.64)];
[y, ifail] = f11xn(trans, a, irow, icol, check, x)
```

```
y =
  -0.7900 + 1.4500i
  -0.2500 - 0.5700i
   3.8200 + 2.2600i
  -3.2800 - 3.7300i
   1.1600 - 0.7800i
ifail =
           0
```