F11JRF – NAG Fortran Library Routine Document

Note. Before using this routine, please read the Users' Note for your implementation to check the interpretation of bold italicised terms and other implementation-dependent details.

1 Purpose

F11JRF solves a system of linear equations involving the preconditioning matrix corresponding to SSOR applied to a complex sparse Hermitian matrix, represented in symmetric coordinate storage format.

2 Specification

```
SUBROUTINE F11JRF(N, NNZ, A, IROW, ICOL, RDIAG, OMEGA, CHECK, Y,
1                 X, IWORK, IFAIL)
INTEGER           N, NNZ, IROW(NNZ), ICOL(NNZ), IWORK(N+1), IFAIL
complex           A(NNZ), Y(N), X(N)
real              RDIAG(N), OMEGA
CHARACTER*1       CHECK
```

3 Description

This routine solves a system of equations:

$$Mx = y$$

involving the preconditioning matrix:

$$M = \frac{1}{\omega(2-\omega)}(D+\omega L)D^{-1}(D+\omega L)^H$$

corresponding to symmetric successive-over-relaxation (SSOR) [1] on a linear system $Ax = b$, where A is a sparse complex Hermitian matrix stored in symmetric coordinate storage (SCS) format (see Section 2.1.2 of the Chapter Introduction).

In the definition of M given above D is the diagonal part of A, L is the strictly lower triangular part of A and ω is a user-defined relaxation parameter. Note that since A is Hermitian the matrix D is necessarily real.

4 References

[1] Young D (1971) *Iterative Solution of Large Linear Systems* Academic Press, New York

5 Parameters

1: N — INTEGER *Input*

On entry: n, the order of the matrix A.

Constraint: N $\geq$ 1.

2: NNZ — INTEGER *Input*

On entry: the number of non-zero elements in the lower triangular part of the matrix A.

Constraint: 1 $\leq$ NNZ $\leq$ N $\times$ (N+1)/2.

3: A(NNZ) — ***complex*** array *Input*

On entry: the non-zero elements in the lower triangular part of 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 routine F11ZPF may be used to order the elements in this way.

4: IROW(NNZ) — INTEGER array *Input*

5: ICOL(NNZ) — INTEGER array *Input*

On entry: the row and column indices of the non-zero elements supplied in A.

Constraints: IROW and ICOL must satisfy the following constraints (which may be imposed by a call to F11ZPF):

$1 \leq \text{IROW}(i) \leq \text{N}$, $1 \leq \text{ICOL}(i) \leq \text{IROW}(i)$, for $i = 1, 2, \ldots,\text{NNZ}$.
$\text{IROW}(i-1) < \text{IROW}(i)$, or
$\text{IROW}(i-1) = \text{IROW}(i)$ and $\text{ICOL}(i-1) < \text{ICOL}(i)$, for $i = 2, 3, \ldots,\text{NNZ}$.

6: RDIAG(N) — ***real*** array *Input*

On entry: the elements of the diagonal matrix D^{-1}, where D is the diagonal part of A. Note that since A is Hermitian the elements of D^{-1} are necessarily real.

7: OMEGA — ***real*** *Input*

On entry: the relaxation parameter ω.

Constraint: $0.0 \leq \text{OMEGA} \leq 2.0$.

8: CHECK — CHARACTER*1 *Input*

On entry: specifies whether or not the input data should be checked:

if CHECK = 'C', checks are carried out on the values of N, NNZ, IROW, ICOL and OMEGA;
if CHECK = 'N', none of these checks are carried out.

Constraint: CHECK = 'C' or 'N'.

9: Y(N) — ***complex*** array *Input*

On entry: the right-hand side vector y.

10: X(N) — ***complex*** array *Output*

On exit: the solution vector x.

11: IWORK(N+1) — INTEGER array *Workspace*

12: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Errors and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

IFAIL = 1

On entry, CHECK ≠ 'C' or 'N'.

IFAIL = 2

On entry, N < 1,
or NNZ < 1,
or NNZ > N × (N+1)/2,
or OMEGA lies outside the interval [0.0,2.0].

IFAIL = 3

On entry, the arrays IROW and ICOL fail to satisfy the following constraints:

$1 \leq \text{IROW}(i) \leq \text{N}$ and $1 \leq \text{ICOL}(i) \leq \text{IROW}(i)$, for $i = 1, 2, \ldots, \text{NNZ}$.
$\text{IROW}(i-1) < \text{IROW}(i)$, or
$\text{IROW}(i-1) = \text{IROW}(i)$ and $\text{ICOL}(i-1) < \text{ICOL}(i)$, for $i = 2, 3, \ldots, \text{NNZ}$.

Therefore a non-zero element has been supplied which does not lie in the lower triangular part of A, is out of order, or has duplicate row and column indices. Call F11ZPF to reorder and sum or remove duplicates.

IFAIL = 4

On entry, a row of A has no diagonal entry.

7 Accuracy

The computed solution x is the exact solution of a perturbed system of equations $(M + \delta M)x = y$, where

$$|\delta M| \leq c(n)\epsilon |D + \omega L||D^{-1}||(D + \omega L)^T|,$$

$c(n)$ 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 F11JRF is proportional to NNZ.

9 Example

None.