NAG Toolbox for MATLAB

f07jv

1 Purpose

f07jv computes error bounds and refines the solution to a complex system of linear equations $AX = B$, where A is an n by n Hermitian positive-definite tridiagonal matrix and X and B are n by r matrices, using the modified Cholesky factorization returned by f07jr and an initial solution returned by f07js. Iterative refinement is used to reduce the backward error as much as possible.

2 Syntax

```
[x, ferr, berr, info] = f07jv(uplo, d, e, df, ef, b, x, 'n', n,
'nrhs_p', nrhs_p)
```

3 Description

f07jv should normally be preceded by calls to f07jr and f07js. f07jr computes a modified Cholesky factorization of the matrix A as

$$A = LDL^{\mathrm{H}},$$

where L is a unit lower bidiagonal matrix and D is a diagonal matrix, with positive diagonal elements. f07js then utilizes the factorization to compute a solution, $\hat{X}$, to the required equations. Letting $\hat{x}$ denote a column of $\hat{X}$, f07jv computes a *component-wise backward error*, β, the smallest relative perturbation in each element of A and b such that $\hat{x}$ is the exact solution of a perturbed system

$$(A+E)\hat{x} = b+f, \quad \text{with} \quad |e_{ij}| \leq \beta|a_{ij}|, \quad \text{and} \quad |f_j| \leq \beta|b_j|.$$

The function also estimates a bound for the *component-wise forward error* in the computed solution defined by $\max|x_i - \hat{x}_i|/\max|\hat{x}_i|$, where x is the corresponding column of the exact solution, X.

Note that the modified Cholesky factorization of A can also be expresses as

$$A = U^{\mathrm{H}}DU,$$

where U is unit upper bidiagonal.

4 References

Anderson E, Bai Z, Bischof C, Blackford S, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A and Sorensen D 1999 *LAPACK Users' Guide* (3rd Edition) SIAM, Philadelphia URL: http://www.netlib.org/lapack/lug

5 Parameters

5.1 Compulsory Input Parameters

1: **uplo – string**

Specifies the form of the factorization as follows:

uplo = 'U'

$$A = U^{\mathrm{H}}DU.$$

uplo = 'L'

$$A = LDL^{\mathrm{H}}.$$

Constraint: **uplo** = 'U' or 'L'.

2: **d**$(*)$ **– double array**

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

Must contain the n diagonal elements of the matrix of A.

3: **e**$(*)$ **– complex array**

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

If **uplo** = 'U', **e** must contain the $(n-1)$ superdiagonal elements of the matrix A.

If **uplo** = 'L', **e** must contain the $(n-1)$ subdiagonal elements of the matrix A.

4: **df**$(*)$ **– double array**

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

Must contain the n diagonal elements of the diagonal matrix D from the LDL^{T} factorization of A.

5: **ef**$(*)$ **– complex array**

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

If **uplo** = 'U', **ef** must contain the $(n-1)$ superdiagonal elements of the unit upper bidiagonal matrix U from the $U^{\mathrm{H}}DU$ factorization of A.

If **uplo** = 'L', **ef** must contain the $(n-1)$ subdiagonal elements of the unit lower bidiagonal matrix L from the LDL^{H} factorization of A.

6: **b**(**ldb**,$*$) **– complex array**

The first dimension of the array **b** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{nrhs\_p})$

The n by r matrix of right-hand sides B.

7: **x**(**ldx**,$*$) **– complex array**

The first dimension of the array **x** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{nrhs\_p})$

The n by r initial solution matrix X.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the array **d** The dimension of the array **df**.

n, the order of the matrix A.

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

2: **nrhs_p – int32 scalar**

Default: The second dimension of the array **b** The second dimension of the array **x**.

r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $\mathbf{nrhs\_p} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldb, ldx, work, rwork

5.4 Output Parameters

1: **x(ldx,$*$) – complex array**

The first dimension of the array **x** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{nrhs\_p})$

The n by r refined solution matrix X.

2: **ferr($*$) – double array**

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

Estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \mathbf{ferr}(j)$, where $\hat{x}_j$ is the jth column of the computed solution returned in the array **x** and x_j is the corresponding column of the exact solution X. The estimate is almost always a slight overestimate of the true error.

3: **berr($*$) – double array**

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

Estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

4: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **uplo**, 2: **n**, 3: **nrhs_p**, 4: **d**, 5: **e**, 6: **df**, 7: **ef**, 8: **b**, 9: **ldb**, 10: **x**, 11: **ldx**, 12: **ferr**, 13: **berr**, 14: **work**, 15: **rwork**, 16: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

7 Accuracy

The computed solution for a single right-hand side, $\hat{x}$, satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_\infty = O(\epsilon)\|A\|_\infty$$

and ϵ is the ***machine precision***. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_\infty}{\|x\|_\infty} \leq \kappa(A)\frac{\|E\|_\infty}{\|A\|_\infty},$$

where $\kappa(A) = \|A^{-1}\|_\infty \|A\|_\infty$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* 1999 for further details.

Function f07ju can be used to compute the condition number of A.

8 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ is proportional to nr. At most five steps of iterative refinement are performed, but usually only one or two steps are required.

The real analogue of this function is f07jh.

9 Example

```
uplo = 'Lower';
d = [16;
     41;
     46;
     21];
e = [complex(16, +16);
     complex(18, -9);
     complex(1, -4)];
df = [16;
     9;
     1;
     4];
ef = [complex(1, +1);
     complex(2, -1);
     complex(1, -4)];
b = [complex(64, +16), complex(-16, -32);
     complex(93, +62), complex(61, -66);
     complex(78, -80), complex(71, -74);
     complex(14, -27), complex(35, +15)];
x = [complex(2, +1), complex(-3, -2);
     complex(1, +1), complex(1, +1);
     complex(1, -2), complex(1, -2);
     complex(1, -1), complex(2, +1)];
[xOut, ferr, berr, info] = f07jv(uplo, d, e, df, ef, b, x)
```

```
xOut =
   2.0000 + 1.0000i  -3.0000 - 2.0000i
   1.0000 + 1.0000i   1.0000 + 1.0000i
   1.0000 - 2.0000i   1.0000 - 2.0000i
   1.0000 - 1.0000i   2.0000 + 1.0000i
ferr =
   1.0e-11 *
    0.9038
    0.6093
berr =
     0
     0
info =
           0
```