

NAG Toolbox for MATLAB

f07fb

1 Purpose

f07fb uses the Cholesky factorization

$$A = U^T U \quad \text{or} \quad A = LL^T$$

to compute the solution to a real system of linear equations

$$AX = B,$$

where A is an n by n symmetric positive-definite matrix and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Syntax

```
[a, af, equed, s, b, x, rcond, ferr, berr, info] = f07fb(fact, uplo, a,
af, equed, s, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07fb performs the following steps:

1. If **fact** = 'E', real diagonal scaling factors, D_S , are computed to equilibrate the system:

$$(D_S A D_S)(D_S^{-1} X) = D_S B.$$

Whether or not the system will be equilibrated depends on the scaling of the matrix A , but if equilibration is used, A is overwritten by $D_S A D_S$ and B by $D_S B$.

2. If **fact** = 'N' or 'E', the Cholesky decomposition is used to factor the matrix A (after equilibration if **fact** = 'E') as $A = U^T U$ if **uplo** = 'U' or $A = LL^T$ if **uplo** = 'L', where U is an upper triangular matrix and L is a lower triangular matrix.
3. If the leading i by i principal minor is not positive-definite, then the function returns with **info** = i . Otherwise, the factored form of A is used to estimate the condition number of the matrix A . If the reciprocal of the condition number is less than *machine precision*, **info** $\geq N + 1$ is returned as a warning, but the function still goes on to solve for X and compute error bounds as described below.
4. The system of equations is solved for X using the factored form of A .
5. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.
6. If equilibration was used, the matrix X is premultiplied by D_S so that it solves the original system before equilibration.

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>

Golub G H and Van Loan C F 1996 *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

Higham N J 2002 *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **fact** – string

Specifies whether or not the factorized form of the matrix A is supplied on entry, and if not, whether the matrix A should be equilibrated before it is factorized.

fact = 'F'

af contains the factorized form of A . If **equed** = 'Y', the matrix A has been equilibrated with scaling factors given by **s**. **af** will not be modified.

fact = 'N'

The matrix A will be copied to **af** and factorized.

fact = 'E'

The matrix A will be equilibrated if necessary, then copied to **af** and factorized.

Constraint: **fact** = 'F', 'N' or 'E'.

2: **uplo** – string

If **uplo** = 'U', the upper triangle of A is stored.

If **uplo** = 'L', the lower triangle of A is stored.

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

3: **a(lda,*)** – double array

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

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

The n by n symmetric matrix A .

If **fact** = 'F' and **equed** = 'Y', **a** must have been equilibrated by the scaling factor in **s** as $D_S A D_S$.

If **uplo** = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If **uplo** = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

4: **af(ldaf,*)** – double array

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

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

If **fact** = 'F', **af** contains the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = L L^T$, in the same storage format as **a**. If **equed** $\neq$ 'N', **af** is the factorized form of the equilibrated matrix $D_S A D_S$.

5: **equed** – string

If **fact** = 'N' or 'E', **equed** need not be set.

If **fact** = 'F', **equed** must specify the form of the equilibration that was performed as follows:

if **equed** = 'N', no equilibration;

if **equed** = 'Y', equilibration was performed, i.e., A has been replaced by $D_S A D_S$.

Constraint: if **fact** = 'F', **equed** = 'N' or 'Y'.

6: **s(*) – double array**

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

If **fact** = 'N' or 'E', **s** need not be set.

If **fact** = 'F' and **equed** = 'Y', **s** must contain the scale factors, D_S , for A ; each element of **s** must be positive.

7: **b(lb,*) – double 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 right-hand side matrix B .

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The second dimension of the array **a** The second dimension of the array **af** The dimension of the array **s**.

n , the number of linear equations, i.e., the order of the matrix A .

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

2: **nrhs_p – int32 scalar**

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

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

lda, ldaf, ldb, ldx, work, iwork

5.4 Output Parameters

1: **a(lda,*) – double array**

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

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

If **fact** = 'F' or 'N', or if **fact** = 'E' and **equed** = 'N', **a** is not modified.

If **fact** = 'E' and **equed** = 'Y', **a** contains $D_S A D_S$.

2: **af(ldaf,*) – double array**

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

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

If **fact** = 'N', **af** returns the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = L L^T$ of the original matrix A .

If **fact** = 'E', **af** returns the triangular factor U or L from the Cholesky factorization $A = U^T U$ or $A = L L^T$ of the equilibrated matrix A (see the description of **a** for the form of the equilibrated matrix).

3: **equed – string**

If **fact** = 'F', **equed** is unchanged from entry.

Otherwise, if **info** ≥ 0 , **equed** specifies the form of the equilibration that was performed as specified above.

4: **s(*)** – double array

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

If **fact** = 'F', **s** is unchanged from entry.

Otherwise, if **info** ≥ 0 and **equed** = 'Y', **s** contains the scale factors, D_S , for A ; each element of **s** is positive.

5: **b(ldb,*)** – double 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 $\max(1, \mathbf{nrhs_p})$

If **equed** = 'N', **b** is not modified.

If **equed** = 'Y', **b** contains $D_S B$.

6: **x(ldx,*)** – double 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 $\max(1, \mathbf{nrhs_p})$

If **info** = 0 or **info** $\geq N + 1$, the n by r solution matrix X to the original system of equations. Note that the arrays A and B are modified on exit if **equed** = 'Y', and the solution to the equilibrated system is $D_S^{-1} X$.

7: **rcond** – double scalar

If **info** ≥ 0 , an estimate of the reciprocal condition number of the matrix A (after equilibration if that is performed), computed as $\mathbf{rcond} = 1 / (\|A\|_1 \|A^{-1}\|_1)$.

8: **ferr(*)** – double array

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

If **info** = 0 or **info** $\geq N + 1$, an 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 j th 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 as reliable as the estimate for **rcond**, and is almost always a slight overestimate of the true error.

9: **berr(*)** – double array

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

If **info** = 0 or **info** $\geq N + 1$, an 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).

10: **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: **fact**, 2: **uplo**, 3: **n**, 4: **nrhs_p**, 5: **a**, 6: **lda**, 7: **af**, 8: **ldaf**, 9: **equed**, 10: **s**, 11: **b**, 12: **ldb**, 13: **x**, 14: **ldx**, 15: **rcond**, 16: **ferr**, 17: **berr**, 18: **work**, 19: **iwork**, 20: **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.

info > 0 and **info** ≤ N

If **info** = i and $i \leq n$, the leading minor of order i of A is not positive-definite, so the factorization could not be completed, and the solution has not been computed. **rcond** = 0 is returned.

info = $N + 1$

U is nonsingular, but **rcond** is less than *machine precision*, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of **rcond** would suggest.

7 Accuracy

For each right-hand side vector b , the computed solution x is the exact solution of a perturbed system of equations $(A + E)x = b$, where

$$|E| \leq c(n)\epsilon|U^T||U|,$$

$c(n)$ is a modest linear function of n , and ϵ is the *machine precision*. See Section 10.1 of Higham 2002 for further details.

If $\hat{x}$ is the true solution, then the computed solution x satisfies a forward error bound of the form

$$\frac{\|x - \hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \text{cond}(A, \hat{x}, b)$$

where $\text{cond}(A, \hat{x}, b) = \| |A^{-1}|(|A||\hat{x}| + |b|) \|_\infty / \|\hat{x}\|_\infty \leq \text{cond}(A) = \| |A^{-1}| |A| \|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the j th column of X , then w_c is returned in **berr**(j) and a bound on $\|x - \hat{x}\|_\infty / \|\hat{x}\|_\infty$ is returned in **ferr**(j). See Section 4.4 of Anderson *et al.* 1999 for further details.

8 Further Comments

The factorization of A requires approximately $\frac{1}{3}n^3$ floating-point operations.

For each right-hand side, computation of the backward error involves a minimum of $4n^2$ floating-point operations. Each step of iterative refinement involves an additional $6n^2$ operations. At most five steps of iterative refinement are performed, but usually only one or two steps are required. Estimating the forward error involves solving a number of systems of equations of the form $Ax = b$; the number is usually 4 or 5 and never more than 11. Each solution involves approximately $2n^2$ operations.

The complex analogue of this function is f07fp.

9 Example

```
fact = 'Equilibration';
uplo = 'Upper';
a = [4.16, -3.12, 0.56000000000000001, -0.1;
     0, 5.03, -0.83, 1.18;
     0, 0, 0.76, 0.34;
     0, 0, 0, 1.18];
af = zeros(4,4);
equed = ' ';
s = zeros(4,1);
```

```

b = [8.699999999999999, 8.300000000000001;
     -13.35, 2.13;
      1.89, 1.61;
     -4.14, 5];
[aOut, afOut, equedOut, sOut, bOut, x, rcond, ferr, berr, info] =
f07fb(fact, uplo, a, af, equed, s, b)

aOut =
    4.1600    -3.1200     0.5600    -0.1000
         0     5.0300    -0.8300     1.1800
         0         0     0.7600     0.3400
         0         0         0     1.1800
afOut =
    2.0396    -1.5297     0.2746    -0.0490
         0     1.6401    -0.2500     0.6737
         0         0     0.7887     0.6617
         0         0         0     0.5347
equedOut =
N
sOut =
    0.4903
    0.4459
    1.1471
    0.9206
bOut =
    8.7000     8.3000
   -13.3500     2.1300
    1.8900     1.6100
   -4.1400     5.0000
x =
    1.0000     4.0000
   -1.0000     3.0000
    2.0000     2.0000
   -3.0000     1.0000
rcond =
    0.0103
ferr =
    1.0e-13 *
    0.2308
    0.2286
berr =
    1.0e-16 *
    0.5743
    0.7892
info =
         0

```