

NAG Toolbox for MATLAB

f07hp

1 Purpose

f07hp uses the Cholesky factorization

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

to compute the solution to a complex system of linear equations

$$AX = B,$$

where A is an n by n Hermitian positive-definite band matrix of bandwidth $(2k_d + 1)$ and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Syntax

```
[ab, afb, equed, s, b, x, rcond, ferr, berr, info] = f07hp(fact, uplo,
kd, ab, afb, equed, s, b, 'n', n, 'nrhs_p', nrhs_p)
```

3 Description

f07hp 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'

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

fact = 'N'

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

fact = 'E'

The matrix A will be equilibrated if necessary, then copied to **afb** 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: **kd** – int32 scalar

k_d , the number of superdiagonals of the matrix A if **uplo** = 'U', or the number of subdiagonals if **uplo** = 'L'.

Constraint: **kd** ≥ 0 .

4: **ab(ldab,*)** – complex array

The first dimension of the array **ab** must be at least **kd** + 1

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

The upper or lower triangle of the Hermitian band matrix A , except if **fact** = 'F' and **equed** = 'Y', in which case **ab** must contain the equilibrated matrix $D_S A D_S$.

The matrix is stored in rows 1 to $k_d + 1$, more precisely,

if **uplo** = 'U', the elements of the upper triangle of A within the band must be stored with element A_{ij} in **ab**($k_d + 1 + i - j, j$) for $\max(1, j - k_d) \leq i \leq j$;

if **uplo** = 'L', the elements of the lower triangle of A within the band must be stored with element A_{ij} in **ab**($1 + i - j, j$) for $j \leq i \leq \min(n, j + k_d)$.

5: **afb(ldafb,*)** – complex array

The first dimension of the array **afb** must be at least **kd** + 1

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

If **fact** = 'F', **afb** contains the triangular factor U or L from the Cholesky factorization $A = U^H U$ or $A = L L^H$ of the band matrix A , in the same storage format as A). If **equed** = 'Y', **afb** is the factorized form of the equilibrated matrix A .

6: **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'.

7: **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.

8: **b(lldb,*)** – **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 right-hand side matrix B .

5.2 Optional Input Parameters

1: **n** – **int32 scalar**

Default: The second dimension of the array **ab** The second dimension of the array **afb** 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

ldab, ldafb, ldb, ldx, work, rwork

5.4 Output Parameters

1: **ab(ldab,*)** – **complex array**

The first dimension of the array **ab** must be at least $\mathbf{kd} + 1$

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

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

2: **afb(ldafb,*)** – **complex array**

The first dimension of the array **afb** must be at least $\mathbf{kd} + 1$

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

If **fact** = 'N', **afb** returns the triangular factor U or L from the Cholesky factorization $A = U^H U$ or $A = LL^H$.

If **fact** = 'E', **afb** returns the triangular factor U or L from the Cholesky factorization $A = U^H U$ or $A = LL^H$ of the equilibrated matrix A (see the description of **ab** 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,*) – 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 $\max(1, \mathbf{nrhs_p})$

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

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

6: **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 $\max(1, \mathbf{nrhs_p})$

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: **kd**, 5: **nrhs_p**, 6: **ab**, 7: **ldab**, 8: **afb**, 9: **ldaafb**, 10: **equed**, 11: **s**, 12: **b**, 13: **ldb**, 14: **x**, 15: **ldx**, 16: **rcond**, 17: **ferr**, 18: **berr**, 19: **work**, 20: **rwork**, 21: **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

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

$c(n)$ 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 $\text{cond}(A, \hat{x}, b) = \frac{\| |A|^{-1}(|A|\|\hat{x}\| + |b|) \|_\infty}{\|\hat{x}\|_\infty} \leq \text{cond}(A) = \frac{\| |A|^{-1} |A| \|_\infty}{1} \leq \kappa_\infty(A)$. 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

When $n \gg k$, the factorization of A requires approximately $4n(k + 1)^2$ floating-point operations, where k is the number of superdiagonals.

For each right-hand side, computation of the backward error involves a minimum of $32nk$ floating-point operations. Each step of iterative refinement involves an additional $48nk$ 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 $16nk$ operations.

The real analogue of this function is f07hb.

9 Example



```

fact = 'Equilibration';
uplo = 'U';
kd = int32(1);
ab = [complex(0, 0), complex(1.08, -1.73), complex(-0.04, +0.29),
      complex(-0.33, +2.24);
      complex(9.39, +0), complex(1.69, +0), complex(2.65, +0),
      complex(2.17, +0)];
afb = complex(zeros(2, 4));
equed = ' ';
s = [0; 0; 0; 0];
b = [complex(-12.42, +68.42), complex(54.3, -56.56);
      complex(-9.93, +0.88), complex(18.32, +4.76);
      complex(-27.3, -0.01), complex(-4.4, +9.97);
      complex(5.31, +23.63), complex(9.43, +1.41)];
[abOut, afbOut, equedOut, sOut, bOut, x, rcond, ferr, berr, info] = ...
    f07hp(fact, uplo, kd, ab, afb, equed, s, b)

```

```

abOut =
    0          1.0800 - 1.7300i  -0.0400 + 0.2900i  -0.3300 +
2.2400i
    9.3900          1.6900          2.6500          2.1700
afbOut =
    0          0.3524 - 0.5646i  -0.0358 + 0.2597i  -0.2054 +
1.3942i
    3.0643          1.1167          1.6066          0.4289
equedOut =
N
sOut =
    0.3263
    0.7692
    0.6143
    0.6788
bOut =
-12.4200 +68.4200i  54.3000 -56.5600i
-9.9300 + 0.8800i  18.3200 + 4.7600i
-27.3000 - 0.0100i  -4.4000 + 9.9700i
 5.3100 +23.6300i   9.4300 + 1.4100i
x =
-1.0000 + 8.0000i   5.0000 - 6.0000i
 2.0000 - 3.0000i   2.0000 + 3.0000i
-4.0000 - 5.0000i  -8.0000 + 4.0000i
 7.0000 + 6.0000i  -1.0000 - 7.0000i
rcond =
    0.0076
ferr =
    1.0e-13 *
    0.3695
    0.3207
berr =
    1.0e-16 *
    0.8851
    0.9363
info =
    0

```