

**f04cb**

*f04cb.1*

See Section 8 for further details.

4: **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  $\max(1, \mathbf{nrhs\_p})$ . To solve the equations  $Ax = b$ , where  $b$  is a single right-hand side, **b** may be supplied as a one-dimensional array with length  $\mathbf{ldb} = \max(1, \mathbf{n})$

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

## 5.2 Optional Input Parameters

1: **n** – **int32 scalar**

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

The number of linear equations  $n$ , 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**.

The number of right-hand sides  $r$ , 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, ldb

## 5.4 Output Parameters

1: **ab(ldab,\*)** – **complex array**

The first dimension of the array **ab** must be at least  $2 \times \mathbf{kl} + \mathbf{ku} + 1$

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

If  $\mathbf{ifail} \geq 0$ , **ab** contains details of the factorization.

The upper triangular band matrix  $U$ , with  $k_l + k_u$  superdiagonals, is stored in rows 1 to  $k_l + k_u + 1$  of the array, and the multipliers used to form the matrix  $L$  are stored in rows  $k_l + k_u + 2$  to  $2k_l + k_u + 1$ .

2: **ipiv(\*)** – **int32 array**

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

If  $\mathbf{ifail} \geq 0$ , the pivot indices that define the permutation matrix  $P$ ; at the  $i$ th step row  $i$  of the matrix was interchanged with row **ipiv**( $i$ ). **ipiv**( $i$ ) =  $i$  indicates a row interchange was not required.

3: **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  $\max(1, \mathbf{nrhs\_p})$ . To solve the equations  $Ax = b$ , where  $b$  is a single right-hand side, **b** may be supplied as a one-dimensional array with length  $\mathbf{ldb} = \max(1, \mathbf{n})$

If  $\mathbf{ifail} = 0$  or  $Np1$ , the  $n$  by  $r$  solution matrix  $X$ .

4: **rcond** – double scalar

If **ifail**  $\geq 0$ , an estimate of the reciprocal of the condition number of the matrix  $A$ , computed as  $\mathbf{rcond} = \left( \|A\|_1 \|A^{-1}\|_1 \right)^{-1}$ .

5: **errbnd** – double scalar

If **ifail** = 0 or  $Np1$ , an estimate of the forward error bound for a computed solution  $\hat{x}$ , such that  $\|\hat{x} - x\|_1 / \|x\|_1 \leq \mathbf{errbnd}$ , where  $\hat{x}$  is a column of the computed solution returned in the array **b** and  $x$  is the corresponding column of the exact solution  $X$ . If **rcond** is less than *machine precision*, then **errbnd** is returned as unity.

6: **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**  $< 0$  and **ifail**  $\neq -999$

If **ifail** =  $-i$ , the  $i$ th argument had an illegal value.

**ifail** =  $-999$

Allocation of memory failed. The double allocatable memory required is **n**, and the *complex\*16* allocatable memory required is  $2 \times \mathbf{n}$ . In this case the factorization and the solution  $X$  have been computed, but **rcond** and **errbnd** have not been computed.

**ifail**  $> 0$  and **ifail**  $\leq N$

If **ifail** =  $i$ ,  $u_{ii}$  is exactly zero. The factorization has been completed, but the factor  $U$  is exactly singular, so the solution could not be computed.

**ifail** =  $N + 1$

**rcond** is less than *machine precision*, so that the matrix  $A$  is numerically singular. A solution to the equations  $AX = B$  has nevertheless been computed.

## 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

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

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

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A) \frac{\|E\|_1}{\|A\|_1},$$

where  $\kappa(A) = \|A^{-1}\|_1 \|A\|_1$ , the condition number of  $A$  with respect to the solution of the linear equations. f04cb uses the approximation  $\|E\|_1 = \epsilon \|A\|_1$  to estimate **errbnd**. See Section 4.4 of Anderson *et al.* 1999 for further details.

## 8 Further Comments

The band storage scheme for the array **ab** is illustrated by the following example, when  $n = 6$ ,  $k_l = 1$ , and  $k_u = 2$ . Storage of the band matrix  $A$  in the array **ab**:

*	*	*	+	+	+
*	*	$a_{13}$	$a_{24}$	$a_{35}$	$a_{46}$
*	$a_{12}$	$a_{23}$	$a_{34}$	$a_{45}$	$a_{56}$
$a_{11}$	$a_{22}$	$a_{33}$	$a_{44}$	$a_{55}$	$a_{66}$
$a_{21}$	$a_{32}$	$a_{43}$	$a_{54}$	$a_{65}$	*

Array elements marked \* need not be set and are not referenced by the function. Array elements marked + need not be set, but are defined on exit from the function and contain the elements  $u_{14}$ ,  $u_{25}$  and  $u_{36}$ .

The total number of floating-point operations required to solve the equations  $AX = B$  depends upon the pivoting required, but if  $n \gg k_l + k_u$  then it is approximately bounded by  $O(nk_l(k_l + k_u))$  for the factorization and  $O(n(2k_l + k_u), r)$  for the solution following the factorization. The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham 2002 for further details.

The real analogue of f04cb is f04bb.

## 9 Example

```

kl = int32(1);
ku = int32(2);
ab = [complex(0, 0), complex(0, 0), complex(0, 0), complex(0, 0);
      complex(0, 0), complex(0, 0), complex(0.97, -2.84), complex(0.59, -
0.48);
      complex(0, 0), complex(-2.05, -0.85), complex(-3.99, +4.01),
complex(3.33, -1.04);
      complex(-1.65, +2.26), complex(-1.48, -1.75), complex(-1.06, +1.94),
complex(-0.46, -1.72);
      complex(0, 6.3), complex(-0.77, 2.83), complex(4.48, -1.09),
complex(0, 0)];
b = [complex(-1.06, +21.5), complex(12.85, +2.84);
     complex(-22.72, -53.9), complex(-70.22, +21.57);
     complex(28.24, -38.6), complex(-20.73, -1.23);
     complex(-34.56, +16.73), complex(26.01, +31.97)];
[abOut, ipiv, bOut, rcond, errbnd, ifail] = f04cb(kl, ku, ab, b)

abOut =
    0                0                0                0.5900 -
0.4800i
    0                0                -3.9900 + 4.0100i    3.3300 -
1.0400i
    0                -1.4800 - 1.7500i    -1.0600 + 1.9400i    -1.7692 -
1.8587i
    0 + 6.3000i    -0.7700 + 2.8300i    4.9303 - 3.0086i    0.4338 +
0.1233i
    0.3587 + 0.2619i    0.2314 + 0.6358i    0.7604 + 0.2429i    0
ipiv =
    2
    3
    3
    4
bOut =
   -3.0000 + 2.0000i    1.0000 + 6.0000i
    1.0000 - 7.0000i   -7.0000 - 4.0000i
   -5.0000 + 4.0000i    3.0000 + 5.0000i
    6.0000 - 8.0000i   -8.0000 + 2.0000i
rcond =
    0.0096
errbnd =
    1.1583e-14
ifail =

```

0

---