F04EBFP

NAG Parallel Library Routine Document

Note: before using this routine, please read the Users' Note for your implementation to check for implementation-dependent details. You are advised to enclose any calls to NAG Parallel Library routines between calls to Z01AAFP and Z01ABFP.

1 Description

F04EBFP calculates the solution of a set of real linear equations

$$AX = B \quad \text{or} \quad A^T X = B,$$

with multiple right-hand sides, using an LU factorization, where A and B are n by n and n by r matrices respectively.

The routine first computes an LU factorization of A as $A = PLU$, where P is a permutation matrix, L is lower triangular with unit diagonal elements and U is upper triangular. The routine uses partial pivoting, with row interchanges. An approximation to X is found by forward and backward substitution.

2 Specification

```
      SUBROUTINE F04EBFP(ICNTXT, TRANS, N, NB, A, LDA, NRHS, B, LDB,
     1                   IPIV, IFAIL)
      DOUBLE PRECISION   A(LDA,*), B(LDB,*)
      INTEGER            ICNTXT, N, NB, LDA, NRHS, LDB, IPIV(*), IFAIL
      CHARACTER*1        TRANS
```

3 Usage

3.1 Definitions

The following definitions are used in describing the data distribution within this document:

m_p	–	the number of rows in the Library Grid.
n_p	–	the number of columns in the Library Grid.
p_r	–	the row grid coordinate of the calling processor.
p_c	–	the column grid coordinate of the calling processor.
N_b	–	the blocking factor for the distribution of the rows and columns of the matrix.
numroc(α,b_ℓ,q,s,k)	–	a function which gives the **num**ber of **r**ows **o**r **c**olumns of a distributed matrix owned by the processor with the row or column coordinate q (p_r or p_c), where α is the total number of rows or columns of the matrix, b_ℓ is the blocking factor used (N_b), s is the row or column coordinate of the processor that possesses the first row or column of the distributed matrix and k is either m_p or n_p. The Library provides the function Z01CAFP (NUMROC) for the evaluation of this function.

3.2 Global and Local Arguments

The following global **input** arguments must have the same value on entry to the routine on each processor and the global **output** arguments will have the same value on exit from the routine on each processor:

Global input arguments: TRANS, N, NB, NRHS, IFAIL

Global output arguments: IFAIL

The remaining arguments are local.

3.3 Distribution Strategy

The matrix A must be partitioned into N_b by N_b square blocks and stored in an array A in a cyclic two-dimensional block distribution. In this routine, the logical processor {0,0} of the processor grid must always possess the first block of the distributed matrix (i.e., $s = 0$ in the function numroc). This data distribution is described in more detail in the F04 Chapter Introduction. The right-hand sides of the equation, B, must be stored in the array B, also in a cyclic two-dimensional block distribution.

3.4 Related Routines

This routine assumes that the data has already been correctly distributed, and if this is not the case will fail to produce correct results. The Library provides many support routines for the generation, scattering/gathering and input/output of matrices/vectors in cyclic two-dimensional block form. The following routines may be used in conjunction with F04EBFP:

Real matrix generation:	F01ZSFP
Real matrix input:	X04BGFP
Real matrix output:	X04BHFP

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP.

Note: the value of ICNTXT **must not** be changed.

2: TRANS — CHARACTER*1 *Global Input*

On entry: indicates the form of the equations as follows:

if TRANS = 'N', then $AX = B$ is solved for X;
if TRANS = 'T' or 'C', then $A^T X = B$ is solved for X.

Constraint: TRANS = 'N', 'T' or 'C'.

3: N — INTEGER *Global Input*

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

Constraint: N $\geq$ 0.

4: NB — INTEGER *Global Input*

On entry: N_b, the blocking factor used to distribute the rows and columns of the matrices A and B.

Constraints: NB $\geq$ 1.

5: A(LDA,$*$) — DOUBLE PRECISION array *Local Input/Local Output*

Note: the size of the second dimension of the array A must be at least max(1,numroc(N,NB,p_c,0,n_p)).

On entry: the local part of the matrix A.

On exit: A is overwritten by the factors L and U distributed in the same cyclic two-dimensional block fashion; the unit diagonal elements of L are not stored.

6: LDA — INTEGER *Local Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F04EBFP is called.

Constraint: LDA $\geq$ max(1,numroc(N,NB,p_r,0,m_p)).

7: NRHS — INTEGER *Global Input*

On entry: r, the number of right-hand sides.

Constraint: NRHS ≥ 0.

8: B(LDB,$*$) — DOUBLE PRECISION array *Local Input/Local Output*

Note: the size of the second dimension of the array B must be at least max(1,numroc(NRHS,NB,p_c,0,n_p)).

On entry: the local part of the the n by r right-hand side matrix B.

On exit: the n by r solution matrix X distributed in the same cyclic two-dimensional block distribution.

9: LDB — INTEGER *Local Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F04EBFP is called.

Constraint: LDB $\geq$ max(1,numroc(N,NB,p_r,0,m_p)).

10: IPIV($*$) — INTEGER array *Local Output*

Note: the dimension of the array IPIV must be at least NB + numroc(N,NB,p_r,0,m_p).

On exit: contains the pivot indices. The global row IPIV(k) of the matrix A was interchanged with the local row k. This array is aligned with the distributed matrix A.

11: IFAIL — INTEGER *Global Input/Global Output*

The NAG Parallel Library provides a mechanism, via the routine Z02EAFP, to reduce the amount of parameter validation performed by this routine. For a full description refer to the Z02 Chapter Introduction.

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this argument (described in the Essential Introduction) the recommended values are:

IFAIL = 0, if multigridding is **not** employed;
IFAIL = -1, if multigridding is employed.

On exit: IFAIL = 0 (or -9999 if reduced error checking is enabled) unless the routine detects an error (see Section 5).

5 Errors and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output from the root processor (or processor {0,0} when the root processor is not available) on the current error message unit (as defined by X04AAF).

5.1 Full Error Checking Mode Only

IFAIL = -2000

The routine has been called with an invalid value of ICNTXT on one or more processors.

IFAIL = -1000

The logical processor grid and library mechanism (Library Grid) have not been correctly defined, see Z01AAFP.

IFAIL = $-i$

On entry, the ith argument had an invalid value. This error occurred either because a global argument did not have the same value on all the logical processors (see Section 3.2), or because its value was incorrect. An explanatory message distinguishes between these two cases.

5.2 Any Error Checking Mode

IFAIL = 1

A diagonal element u_{ii} of U is exactly zero. The factorization has been completed but the factor U is exactly singular, and division by zero will occur if it is used to solve a system of linear equations.

6 Further Comments

The total number of floating-point operations is approximately $\frac{2}{3}n^3 + 2n^2r$.

6.1 Algorithmic Detail

The routine uses a block-partitioned LU factorization with partial pivoting. See Anderson *et al.* [2] for further details. To compute X, forward and backward substitution are used.

6.2 Parallelism Detail

Each processor column performs an LU factorization with partial pivoting on successive column blocks of the matrix. Details of this factorization and pivoting are passed to all processors which perform the update of the remaining matrix in parallel. Forward and backward substitution are applied to each column block of the right-hand sides in parallel.

6.3 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

$$\|E\| \leq \epsilon g c(n)\|A\|,$$

$c(n)$ is a modest function of n, ϵ is the ***machine precision*** and g is $\max|u_{ij}|$.

If $\hat{x}$ is the true solution, then the computed solution x satisfies the bound

$$\frac{\|x-\hat{x}\|}{\|x\|} \leq gc(n)\mathrm{cond}(A)\epsilon$$

where $\mathrm{cond}(A) = \|A\| \cdot \|A^{-1}\|$.

7 References

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

[2] 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

8 Example

To solve the system of equations $AX = B$, where

$$A = \begin{pmatrix} 1.80 & 2.88 & 2.05 & -0.89 \\ 5.25 & -2.95 & -0.95 & -3.80 \\ 1.58 & -2.69 & -2.90 & -1.04 \\ -1.11 & -0.66 & -0.59 & 0.80 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 9.52 & 18.47 \\ 24.35 & 2.25 \\ 0.77 & -13.28 \\ -6.22 & -6.21 \end{pmatrix}.$$

The example uses a 2 by 2 logical processor grid and a block size of 2 for both A and B.

Note: the listing of the Example Program presented below does not give a full pathname for the data file being opened, but in general the user must give the full pathname in this and any other OPEN statement.

8.1 Example Text

```
*     F04EBFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NB
      PARAMETER        (NB=2)
      INTEGER          NMAX, LDA, LDB, NRHMAX, LW
      PARAMETER        (NMAX=8,LDA=NMAX,LDB=NMAX,NRHMAX=2,LW=NMAX)
*     .. Local Scalars ..
      INTEGER          ICNTXT, IFAIL, MP, N, NP, NRHS
      LOGICAL          ROOT
      CHARACTER        TRANS
      CHARACTER*80     FORMAT
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NRHMAX), WORK(LW)
      INTEGER          IPIV(NMAX+NB)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F04EBFP, X04BGFP, X04BHFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'F04EBFP Example Program Results'
*
      MP = 2
      NP = 2
      IFAIL = 0
*
*     Set up Library Grid
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      OPEN (NIN,FILE='f04ebfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NRHS, FORMAT
*
      IF (N.LE.NMAX .AND. NRHS.LE.NRHMAX) THEN
*
         IFAIL = 0
*
*        Read in matrices A and B
*
         CALL X04BGFP(ICNTXT,NIN,N,N,NB,A,LDA,IFAIL)
*
         CALL X04BGFP(ICNTXT,NIN,N,NRHS,NB,B,LDB,IFAIL)
*
         TRANS = 'N'
*
         CALL F04EBFP(ICNTXT,TRANS,N,NB,A,LDA,NRHS,B,LDB,IPIV,IFAIL)
*
*        Print solution(s)
*
         IF (ROOT) THEN
```

```
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Solution(s)'
            WRITE (NOUT,*)
         END IF
*
         CALL X04BHFP(ICNTXT,NOUT,N,NRHS,NB,B,LDB,FORMAT,WORK,IFAIL)
*
      END IF
*
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
```

8.2 Example Data

```
F04EBFP Example Program Data
  4 2 '(4F12.4)'               :Values of N, NRHS and FORMAT
  1.80   2.88   2.05  -0.89
  5.25  -2.95  -0.95  -3.80
  1.58  -2.69  -2.90  -1.04
 -1.11  -0.66  -0.59   0.80   :End of matrix A
  9.52  18.47
 24.35   2.25
  0.77 -13.28
 -6.22  -6.21                 :End of matrix B
```

8.3 Example Results

```
 F04EBFP Example Program Results

 Solution(s)

      1.0000      3.0000
     -1.0000      2.0000
      3.0000      4.0000
     -5.0000      1.0000
```