F04GBFP

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

F04GBFP solves a set of real linear least-squares problems, posed in the form:

$$\text{find} \quad x^{(i)} \text{ to minimize } \|Ax^{(i)} - b^{(i)}\|_2 \quad \text{for} \quad i = 1, 2, ..., r.$$

Here A is an m by n matrix with $m \geq n$ and rank n. The right hand side vectors $b^{(i)}$ have m elements, and the solution vectors $x^{(i)}$ have n elements. The above problem is also referred to as solving an **overdetermined system of linear equations**.

The routine first computes a QR factorization of A, $A = Q\begin{pmatrix} R \\ 0 \end{pmatrix}$ (see Section 6.1), where Q is an m by m orthogonal matrix and R is an n by n upper triangular matrix. Then the right-hand side is transformed by applying Q^T from the left. The resulting upper triangular system is solved using a back-substitution algorithm.

2 Specification

```
 SUBROUTINE F04GBFP(ICNTXT, M, N, NB, A, LDA, NRHS, B, LDB, STDERR,
1                   WORK, LW, IFAIL)
 DOUBLE PRECISION   A(LDA,*), B(LDB,*), STDERR(NRHS), WORK(LW)
 INTEGER            ICNTXT, M, N, NB, LDA, NRHS, LDB, LW, IFAIL
```

3 Data Distribution

3.1 Definitions

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

m_p	–	the number of rows in the logical processor grid.
n_p	–	the number of columns in the logical processor 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$(\alpha, b_\ell, q, s, k)$	–	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 n_p or m_p. The Library provides the function Z01CAFP (NUMROC) for the evaluation of this function.

3.2 Global and Local Arguments

The input arguments M, N, NB, NRHS and IFAIL are all global and so must have the same values on entry to the routine on each processor. The output arguments STDERR and IFAIL are global, and will have the same value on exit from the routine on each processor. 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 2-d 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 2-d block distribution.

This routine assumes that the data has already been correctly distributed, and if this is not the case will fail to produce correct results. However, the Library provides some utility routines which assist you in distributing data correctly. Descriptions of these routines can be found in Chapters F01 and X04 of the NAG Parallel Library Manual.

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the BLACS context used by the communication mechanism, usually returned by a call to Z01AAFP.

2: M — INTEGER *Global Input*

On entry: the number of rows of the matrix A, m.

Constraint: M $\geq$ 1.

3: N — INTEGER *Global Input*

On entry: the number of columns of the matrix A, n.

Constraint: M $\geq$ N $\geq$ 1.

4: NB — INTEGER *Global Input*

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

Constraints: NB $\geq$ 1.

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

Note: 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 m by n matrix A to be factorized.

On exit: the elements below the diagonal of A are used as workspace and the upper triangle is overwritten by the upper triangular matrix R.

6: LDA — INTEGER *Local Input*

On entry: the size of the first dimension of the array A as declared in the (sub)program from which F04GBFP is called.

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

7: NRHS — INTEGER *Global Input*

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

Constraint: NRHS $\geq$ 1.

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

Note: 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 r right-hand sides $b^{(i)}$, for $i = 1, 2, ..., r$ distributed in cyclic 2-d block form.

On exit: the n by r solution matrix $x_j^{(i)}$, for $j = 1, 2, ..., n$; $i = 1, 2, ..., r$ (stored in the first n rows) distributed in the same cyclic 2-d block distribution. The remaining $m - n$ rows contain the vectors $c_2^{(i)}$, for $i = 1, 2, \ldots, r$ (as described in Section 6.1) in the cyclic 2-d block distribution.

9: LDB — INTEGER *Local Input*

On entry: the size of the first dimension of the array B as declared in the (sub)program from which F04GBFP is called.

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

10: STDERR(NRHS) — DOUBLE PRECISION array *Global Output*

On exit: the standard error of the solution vectors $x^{(i)}$, for $i = 1, 2, \ldots, r$, defined as $\|b^{(i)} - Ax^{(i)}\|_2/\sqrt{m-n}$ if $m > n$, and zero if $m = n$.

11: WORK(LW) — DOUBLE PRECISION array *Local Workspace*

12: LW — INTEGER *Local Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F04GBFP is called.

Constraint: LW $\geq$ NB^2 + $(d_1 + d_2)$NB + d_2, where

d_1 = numroc(M,NB,p_r,0,m_p);
d_2 = numroc(N,NB,p_c,0,n_p).

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

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (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 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 on the current error message unit (as defined by X04AAF).

Errors detected by the routine:

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.

IFAIL = 1

The rank of A is less than n; a diagonal element of R detected to be zero.

6 Further Comments

6.1 Algorithmic Detail

For an m by n matrix A ($m \geq n$), the factorization is given by:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix}$$

where R is an n by n upper triangular matrix and Q is an m by m orthogonal matrix. It is sometimes more convenient to write the factorization as

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix}$$

which reduces to

$$A = Q_1 R,$$

where Q_1 consists of the first n columns of Q, and Q_2 the remaining $m - n$ columns.

The matrix Q is not formed explicitly but is represented as a product of n elementary reflectors. See [2] for details of the block method used by the routine. This factorisation allows the solution of the linear least-squares problem, since

$$\|b - Ax\|_2 = \|Q^T b - Q^T Ax\|_2 = \left\| \begin{matrix} c_1 - Rx \\ c_2 \end{matrix} \right\|_2,$$

where

$$c \equiv \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} Q_1^T b \\ Q_2^T b \end{pmatrix} = Q^T b$$

and c_1 is an n-element vector. Then x is the solution of the upper triangular system

$$Rx = c_1.$$

The residual vector r is given by

$$r = b - Ax = Q \begin{pmatrix} 0 \\ c_2 \end{pmatrix}.$$

The residual sum of squares $\|r\|_2^2$ may be computed without forming r explicitly, since

$$\|r\|_2 = \|b - Ax\|_2 = \|c_2\|_2.$$

Information on the sensitivity of the least-squares problem can be found in [1].

6.2 Parallelism Detail

The Level 3 BLAS operations are carried out in parallel within the routine.

6.3 Accuracy

The computed factorization is the exact factorization of a nearby matrix $A + E$, where

$$\|E\|_2 = \epsilon c(m, n)\|A\|_2,$$

ϵ is ***machine precision*** and $c(m, n)$ is a modest function of m and n.

7 References

[1] Golub G H and Van Loan C F (1989) *Matrix Computations* Johns Hopkins University Press (2nd Edition), Baltimore

[2] Anderson E, Bai Z, Bischof C, Demmel J, Dongarra J J, Du Croz J J, Greenbaum A, Hammarling S, McKenney A, Ostrouchov S and Sorensen D (1995) *LAPACK Users' Guide* (2nd Edition) SIAM, Philadelphia

8 Example

To solve the pair of linear least-squares problem

$$\min \|Ax^{(i)} - b^{(i)}\|_2 \quad \text{for} \quad i = 1,2$$

where $b^{(1)}$ and $b^{(2)}$ are the columns of the matrix B,

$$A = \begin{pmatrix} -0.57 & -1.28 & -0.39 & 0.25 \\ -1.93 & 1.08 & -0.31 & -2.14 \\ 2.30 & 0.24 & 0.40 & -0.35 \\ -1.93 & 0.64 & -0.66 & 0.08 \\ 0.15 & 0.30 & 0.15 & -2.13 \\ -0.02 & 1.03 & -1.43 & 0.50 \end{pmatrix} \text{ and } \quad B = \begin{pmatrix} -3.15 & 2.19 \\ -0.11 & -3.64 \\ 1.99 & 0.57 \\ -2.70 & 8.23 \\ 0.26 & -6.35 \\ 4.50 & -1.48 \end{pmatrix}.$$

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

```
*     F04GBFP 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          MMAX, NMAX, LDA, LDB, NRHMAX, LW
      PARAMETER        (MMAX=8,NMAX=6,LDA=MMAX,LDB=MMAX,NRHMAX=2,
     +                 LW=NB*NB+(MMAX*NMAX)*NB+NMAX)
*     .. Local Scalars ..
      INTEGER          I, ICNTXT, IFAIL, M, MP, N, NP, NRHS
      LOGICAL          ROOT
      CHARACTER*80     FORMAT
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), B(LDB,NRHMAX), STDERR(NRHMAX),
     +                 WORK(LW)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F04GBFP, X04BGFP, X04BHFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'F04GBFP Example Program Results'
*
      MP = 2
      NP = 2
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      OPEN (NIN,FILE='f04gbfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) M, N, NRHS, FORMAT
*
      IF (M.LE.MMAX .AND. N.LE.NMAX .AND. NRHS.LE.NRHMAX) THEN
```

```
*
         IFAIL = 0
*
*        Read in matrices A and B
*
         CALL X04BGFP(ICNTXT,NIN,M,N,NB,A,LDA,IFAIL)
*
         CALL X04BGFP(ICNTXT,NIN,M,NRHS,NB,B,LDB,IFAIL)
*
         CALL F04GBFP(ICNTXT,M,N,NB,A,LDA,NRHS,B,LDB,STDERR,WORK,LW,
     +                IFAIL)
*
*        Print solution(s)
*
         IF (ROOT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Least-squares solution(s)'
            WRITE (NOUT,*)
         END IF
*
         CALL X04BHFP(ICNTXT,NOUT,N,NRHS,NB,B,LDB,FORMAT,WORK,IFAIL)
*
         IF (ROOT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Standard error(s)'
            WRITE (NOUT,*)
            DO 20 I = 1, NRHS
               WRITE (NOUT,99999) I, STDERR(I)
   20       CONTINUE
         END IF

      END IF
*
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
*
99999 FORMAT (1X,I4,E16.2)
      END
```

8.2 Example Data

```
F04GBFP Example Program Data
 6 4 2 '(4F12.4)'       :Values of M, N, NRHS and FORMAT
-0.57 -1.28 -0.39  0.25
-1.93  1.08 -0.31 -2.14
 2.30  0.24  0.40 -0.35
-1.93  0.64 -0.66  0.08
 0.15  0.30  0.15 -2.13
-0.02  1.03 -1.43  0.50  :End of matrix A
-3.15  2.19
-0.11 -3.64
 1.99  0.57
-2.70  8.23
 0.26 -6.35
```

```
4.50 -1.48              :End of matrix B
```

8.3 Example Results

```
F04GBFP Example Program Results

 Least-squares solution(s)

     1.5146     -1.5838
     1.8621      0.5536
    -1.4467      1.3491
     0.0396      2.9600

 Standard error(s)

   1        0.18E+01
   2        0.53E+01
```