F07TGFP (PDTRCON)

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

F07TGFP (PDTRCON) estimates the condition number of a real triangular matrix A_s, in either the 1-norm or the ∞-norm:

$$\kappa_1(A_s) = ||A_s||_1||A_s^{-1}||_1 \quad \text{or} \quad \kappa_\infty(A_s) = ||A_s||_\infty||A_s^{-1}||_\infty,$$

distributed on a logical grid of processors in a cyclic two-dimensional block distribution; A_s is a submatrix of a larger m_A by n_A matrix A, i.e.,

$$A_s(1:m,1:n) \equiv A(i_A:i_A+m-1, j_A:j_A+n-1).$$

Note: if $i_A = j_A = 1$, $m = m_A$ and $n = n_A$, then $A_s = A$.

Because the condition number is infinite if A_s is singular, the routine actually returns an estimate of the ***reciprocal*** of the condition number.

The routine computes $||A_s||_1$ or $||A_s||_\infty$ exactly and uses Higham's implementation of Hager's method (see, [2]) to estimate $||A_s^{-1}||_1$ or $||A_s^{-1}||_\infty$. Note that $\kappa_\infty(A_s) = \kappa_1(A_s^T)$.

2 Specification

```
 SUBROUTINE F07TGFP(NORM, UPLO, DIAG, N, A, IA, JA, IDESCA, RCOND,
1                   WORK, LWORK, IWORK, LIWORK, INFO)
 ENTRY      PDTRCON(NORM, UPLO, DIAG, N, A, IA, JA, IDESCA, RCOND,
1                   WORK, LWORK, IWORK, LIWORK, INFO)
 DOUBLE PRECISION   A(*), RCOND, WORK(LWORK)
 INTEGER            N, IA, JA, IDESCA(*), LWORK, IWORK(LIWORK),
1                   LIWORK, INFO
 CHARACTER*1        NORM, UPLO, DIAG
```

The ENTRY statement enables the routine to be called by its ScaLAPACK name.

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.
M_b^X	–	the blocking factor for the distribution of the rows of a matrix X.
N_b^X	–	the blocking factor for the distribution of the columns of a matrix X.
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 (M_b^X or N_b^X), 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.

indxg2p(i_g,b_ℓ,q,s,k)	–	a function which gives the processor row or column coordinate which possess the row or column index i_g of the distributed full matrix A. The arguments b_ℓ, q, s and k have the same meaning as in the function numroc. The Library provides the function Z01CDFP (INDXG2P) 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:	NORM, UPLO, DIAG, N, IA, JA, IDESCA(1), IDESCA(3:8)
Global output arguments:	RCOND, INFO

The remaining arguments are local.

3.3 Distribution Strategy

The matrix A must be partitioned into M_b^A by N_b^A rectangular blocks (in this release $M_b^A = N_b^A$) and stored in an array A in a cyclic two-dimensional block distribution. This data distribution is described in more detail in the F07 Chapter Introduction.

3.4 Related Routines

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 F07TGFP (PDTRCON):

Real matrix generation:	F01ZQFP
Real matrix input:	X04BCFP
Real matrix output:	X04BDFP
Real matrix gather:	F01WAFP
Real matrix scatter:	F01WNFP

4 Arguments

1: NORM — CHARACTER*1 *Global Input*

On entry: indicates whether $\kappa_1(A_s)$ or $\kappa_\infty(A_s)$ is estimated as follows:

if NORM = '1' or 'O', then $\kappa_1(A_s)$ is estimated;
if NORM = 'I' then $\kappa_\infty(A_s)$ is estimated.

Constraint: NORM = '1','O','I'.

2: UPLO — CHARACTER*1 *Global Input*

On entry: indicates whether A_s is upper or lower triangular as follows:

if UPLO = 'U', then A_s is upper triangular;
if UPLO = 'L', then A_s is lower triangular.

Constraint: UPLO = 'U' or 'L'.

3: DIAG — CHARACTER*1 *Global Input*

On entry: indicates whether A_s is a non-unit or unit triangular matrix as follows:

if DIAG = 'N' then A_s is a non-unit triangular matrix;
if DIAG = 'U' then A_s is a unit triangular matrix; the diagonal elements are not referenced and are assumed to be 1.

Constraint: DIAG = 'N' or 'U'.

4: N — INTEGER *Global Input*

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

Constraint: $0 \leq \text{N} \leq \min(\text{IDESCA}(3), \text{IDESCA}(4))$.

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

Note: array A is formally defined as a vector. However, you may find it more convenient to consider A as a two-dimensional array of dimension (IDESCA(9),γ), where $\gamma \geq$ numroc(JA+N$-$1,IDESCA(6),p_c,IDESCA(8),n_p).

On entry: the local part of the matrix A which may contain parts of the n by n triangular submatrix A_s.

If UPLO = 'U', A_s is upper triangular and the elements of the matrix below the diagonal are not referenced;

if UPLO = 'L', A_s is lower triangular and the elements of the matrix above the diagonal are not referenced.

If DIAG = 'U', the diagonal elements of A_s are not referenced, but are assumed to be 1.

6: IA — INTEGER *Global Input*

On entry: i_A, the row index of matrix A that identifies the first row of the submatrix A_s.

Constraints: $1 \leq \text{IA} \leq \text{IDESCA}(3) - \text{N} + 1$ and mod(IA$-$1,IDESCA(5)) = 0.

7: JA — INTEGER *Global Input*

On entry: j_A, the column index of matrix A that identifies the first column of the submatrix A_s.

Constraints: $1 \leq \text{JA} \leq \text{IDESCA}(4) - \text{N} + 1$ and mod(JA$-$1,IDESCA(6)) = 0.

8: IDESCA($*$) — INTEGER array *Local Input*

Note: the dimension of the array IDESCA must be at least 9.

Distribution: the array elements IDESCA(1) and IDESCA(3),...,IDESCA(8) must be global to the processor grid and the array elements IDESCA(2) and IDESCA(9) are local to each processor.

On entry: the description array for the matrix A. This array must contain details of the distribution of the matrix A and the logical processor grid.

IDESCA(1), the descriptor type. For this routine, which uses a cyclic two-dimensional block distribution, IDESCA(1) = 1;

IDESCA(2), the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP;

IDESCA(3), the number of rows, m_A, of the matrix A;

IDESCA(4), the number of columns, n_A, of the matrix A;

IDESCA(5), the blocking factor, M_b^A, used to distribute the rows of the matrix A;

IDESCA(6), the blocking factor, N_b^A, used to distribute the columns of the matrix A;

IDESCA(7), the processor row index over which the first row of the matrix A is distributed;

IDESCA(8), the processor column index over which the first column of the matrix A is distributed;

IDESCA(9), the leading dimension of the conceptual two-dimensional array A.

Constraints:

IDESCA(1) = 1;
IDESCA(3) $\geq$ 0; IDESCA(4) $\geq$ 0;
IDESCA(5) = IDESCA(6); IDESCA(5) $\geq$ 1; IDESCA(6) $\geq$ 1;
0 $\leq$ IDESCA(7) $\leq m_p - 1$; 0 $\leq$ IDESCA(8) $\leq n_p - 1$;
IDESCA(9) $\geq$ max(1,numroc(IDESCA(3),IDESCA(5),p_r,IDESCA(7),m_p)).

9: RCOND — DOUBLE PRECISION *Global Output*

On exit: an estimate of the reciprocal of the condition number of A. RCOND is set to zero if exact singularity is detected or if the estimate underflows. If RCOND is less than ***machine precision***, then A is singular to working precision.

10: WORK(LWORK) — DOUBLE PRECISION array *Local Workspace/Local Output*

Note: the dimension of WORK must be at least max(1,LWORK). The minimum value of LWORK required to successfully call this routine can be obtained by setting LWORK = −1. The required size is returned in array element WORK(1).

On exit: WORK(1) contains the minimum dimension of the array WORK required to successfully complete the task.

11: LWORK — INTEGER *Local Input*

On entry: either −1 (see WORK) or the dimension of the array WORK required to successfully complete the task. If LWORK is set to −1 on entry this routine simply performs some initial error checking and then, if these checks are successful, calculates the minimum size of LWORK required.

Constraints:

either LWORK = −1,
or LWORK $\geq 2d_1 + d_2+$ max(2, max(IDESCA(6) $\times w_1, d_1$ + IDESCA(6) $\times$ w_2)), where
c_1 = indxg2p(IA, IDESCA(5), p_r, IDESCA(7), m_p),
c_2 = indxg2p(JA, IDESCA(6), p_c, IDESCA(8), n_p),
d_1 = numroc(N + mod(IA − 1, IDESCA(5)), IDESCA(5), p_r, c_1, m_p),
d_2 = numroc(N + mod(JA−1, IDESCA(6)), IDESCA(6), p_c, c_2, n_p),
w_1 = max(1, $\lceil (m_p - 1)/n_p \rceil$),
w_2 = max(1, $\lceil (n_p - 1)/m_p \rceil$).

12: IWORK(LIWORK) — INTEGER array *Local Workspace/Local Output*

Note: the minimum value of LIWORK required to successfully call this routine can be obtained by setting LIWORK = −1. The required size is returned in array element IWORK(1).

On exit: IWORK(1) contains the minimum dimension of array IWORK required to successfully complete the task.

13: LIWORK — INTEGER *Local Input*

On entry: either −1 (see IWORK) or the dimension of the array WORK required to successfully complete the task. If LIWORK is set to −1 on entry this routine simply performs some initial error checking and then, if these checks are successful, calculates the minimum size of LIWORK required.

Constraints:

either LWORK = −1,
or LIWORK $\geq$ numroc(N + mod(IA−1, IDESCA(5)), IDESCA(5), p_r, α, m_p), where
α = indxg2p(IA, IDESCA(5), p_r, IDESCA(7), m_p).

14: INFO — INTEGER *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 exit: INFO = 0 (or −9999 if reduced error checking is enabled) unless the routine detects an error (see Section 5).

5 Errors and Warnings

If INFO $\neq 0$ 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).

INFO < 0

On entry, one of the arguments was invalid:

if the kth argument is a scalar INFO $= -k$;
if the kth argument is an array and its jth element is invalid, INFO $= -(100\times k + j)$.

This error occured either because a global argument did not have the same value on all logical processors, or because its value on one or more processors was incorrect.

6 Further Comments

The routine performs limited checking for high condition number matrices. It is possible for very ill-conditioned matrices that the routine will cause an arithmetic overflow.

6.1 Algorithmic Detail

The routine computes $\|A_s\|_1$ or $\|A_s\|_\infty$ accurately, and uses Higham's implementation of Hager's method to estimate $\|A_s^{-1}\|_1$ or $\|A_s^{-1}\|_\infty$.

6.2 Parallelism Detail

The Level-3 BLAS operations are carried out in parallel.

6.3 Accuracy

The computed estimate RCOND is never less than the true value ρ, and in practice is nearly always less than 10ρ, although examples can be constructed where RCOND is much larger.

7 References

[1] Blackford L S, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, Dongarra J, Hammarling S, Henry G, Petitet A, Stanley K, Walker D and Whaley R C (1997) ScaLAPACK Users' Guide *SIAM* 3600 University City Science Center, Philadelpia, PA 19104-2688, USA. URL: http://www.netlib.org/scalapack/slug/scalapack_slug.html

[2] Higham N J (1988) FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation *ACM Trans. Math. Software* **14** 381–396

8 Example

To estimate the condition number in the 1-norm of the matrix A, where

$$A = \begin{pmatrix} 4.30 & 0.00 & 0.00 & 0.00 \\ -3.96 & -4.87 & 0.00 & 0.00 \\ 0.40 & 0.31 & -8.02 & 0.00 \\ -0.27 & 0.07 & -5.95 & 0.12 \end{pmatrix}.$$

The true condition number in the 1-norm is 116.41.

The example uses a 2 by 2 logical processor grid and a block size of 2.

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

```
*     F07TGFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          DT
      PARAMETER        (DT=1)
      INTEGER          NB
      PARAMETER        (NB=2)
      INTEGER          NMAX, LDA, IAROW, IACOL, RCONST, CCONST, LWORK,
     +                 LIWORK
      PARAMETER        (NMAX=8,LDA=NMAX,IAROW=0,IACOL=0,RCONST=2,
     +                 CCONST=2,LWORK=2*NMAX,LIWORK=NMAX)
      CHARACTER        DIAG, NORM
      PARAMETER        (DIAG='N',NORM='1')
*     .. Local Scalars ..
      DOUBLE PRECISION RCOND
      INTEGER          IA, ICNTXT, IFAIL, INFO, JA, N, NCOLS, NROWS
      LOGICAL          ROOT
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), WORK(LWORK)
      INTEGER          IDESCA(9), IWORK(LIWORK)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF
      LOGICAL          Z01ACFP
      EXTERNAL         X02AJF, Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F07TGFP, X04BCFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..
*
      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'F07TGFP Example Program Results'
*
      NROWS = RCONST
      NCOLS = CCONST
      IFAIL = 0
*
      CALL Z01AAFP(ICNTXT,NROWS,NCOLS,IFAIL)
*
      OPEN (NIN,FILE='f07tgfpe.d')
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, UPLO
*
      IF (N.LE.NMAX) THEN
*
*        Set the array descriptor of A
*
         IA = 1
```

```
         JA = 1
         IDESCA(1) = DT
         IDESCA(2) = ICNTXT
         IDESCA(3) = N
         IDESCA(4) = N
         IDESCA(5) = NB
         IDESCA(6) = NB
         IDESCA(7) = IAROW
         IDESCA(8) = IACOL
         IDESCA(9) = LDA
*
*         Read A from data file
*
         IFAIL = 0
         CALL X04BCFP(NIN,N,N,A,IA,JA,IDESCA,IFAIL)

*
*         Compute the condition number
*
         CALL F07TGFP(NORM,UPLO,DIAG,N,A,IA,JA,IDESCA,RCOND,WORK,LWORK,
     +                IWORK,LIWORK,INFO)
*
*         Print condition number
*
         IF (ROOT .AND. INFO.EQ.0) THEN
            IF (RCOND.GE.X02AJF()) THEN
               WRITE (NOUT,99999)
     +           'Estimate of the condition number =', 1.0D0/RCOND
            ELSE
               WRITE (NOUT,*) 'A is singular to working precision'
            END IF
         END IF
*
      END IF
*
      CLOSE (NIN)
*
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
*
99999 FORMAT (1X,A,1P,D10.2)
      END
```

8.2 Example Data

```
F07TGFP Example Program Data
4 'L'                   :Value of N and UPLO
 4.30   0.0    0.0   0.0
-3.96  -4.87   0.0   0.0
 0.40   0.31  -8.02  0.0
-0.27   0.07  -5.95  0.12    :End of matrix A
```

8.3 Example Results

```
F07TGFP Example Program Results
Estimate of the condition number =  1.16D+02
```