F07JSFP (PZPTTRS)

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

F07JSFP (PZPTTRS) solves an n by n complex tridiagonal Hermitian positive-definite system of linear equations with multiple right-hand sides, $A_sX = B_s$, where A_s is a submatrix of a larger $n \times n_A$ matrix A,

$$A_s(1:n, 1:n) \equiv A(1:n, j_A : j_A + n - 1);$$

and B_s is a $n \times r$ submatrix of a larger m_B by r matrix B,

$$B_s(1:n, 1:r) \equiv B(i_B : i_B + n - 1, 1:r).$$

Note: if $j_A = 1$ and $n = n_A$, then $A_s = A$; if $i_B = 1$ and $n = m_B$, then $B_s = B$.

The tridiagonal matrix A_s must have been previously factorized by a call to F07JRFP (PZPTTRF), which performs a Cholesky factorization returning details of either the factor U or L according to whether the factorization PU^HDUP^T or $PDLDL^HP^T$ was carried out, respectively. F07JSFP (PZTTRS) then solves the tridiagonal system of equations.

2 Specification

```
      SUBROUTINE F07JSFP(UPLO, N, NRHS, D, E, JA, IDESCA, B, IB, IDESCB,
     1                   AF, LAF, WORK, LWORK, INFO)
      ENTRY      PZPTTRS(UPLO, N, NRHS, D, E, JA, IDESCA, B, IB, IDESCB,
     1                   AF, LAF, WORK, LWORK, INFO)
      INTEGER            N, NRHS, JA, IDESCA(*), IB, IDESCB(9), LAF,
     1                   LWORK, INFO
      COMPLEX*16         D(*), E(*), AF(*), WORK(*)
      CHARACTER*1        UPLO
```

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, for this routine $m_p = 1$ or $m_p = p$;
- n_p – the number of columns in the Library Grid, for this routine $n_p = 1$ or $n_p = p$.
- p – $m_p \times n_p$, the total number of processors in the Library Grid.
- 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.

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: UPLO, N, NRHS, JA, IB, some elements of IDESCA and IDESCB (see Section 4 for a description of IDESCA and IDESCB).

Global output arguments: INFO.

The remaining arguments are local.

3.3 Distribution Strategy

The matrix A is represented by two vectors e (off-diagonal elements) and d (diagonal elements), as in F07JRFP (PZPTTRF). The right-hand sides of the equation, B_s are stored in the array B in a row block distribution. **It is important that** $p \times N_b^A \geq \mathrm{mod}(j_A - 1,\ N_b^A) + n$, with $M_b^B = N_b^A$ and $i_B = j_A$. This restriction states that the mapping for matrices must be blocked, due to alignment restriction and reflecting the nature of the **divide and conquer algorithm** as a task-parallel algorithm. This means that no processor may have more than one block of the matrix.

3.4 Related Routines

The Library provides many support routines for the generation/distribution and input/output of data in column or row block form. The following routines may be used in conjunction with F07JSFP (PZPTTRS):

Real and complex matrix generation:	column block distribution: F01ZZFP and F01ZYFP for the distribution of respectively the vectors d and e.
Complex matrix generation:	row block distribution: F01YZFP
Complex matrix output:	row block distribution: X04BZFP

4 Arguments

1: UPLO — CHARACTER*1 *Global Input*

On entry:

if UPLO = 'U', then E must store details of the upper triangular matrix U as returned by F07JRFP (PZPTTRF);
if UPLO = 'L', then E must store dteails of the lower triangular matrix L as returned by F07JRFP (PZPTTRF).

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

2: N — INTEGER *Global Input*

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

Constraint: N $\geq$ 0.

3: NRHS — INTEGER *Global Input*

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

Constraint: NRHS $\geq$ 1.

4: D($*$) — COMPLEX*16 array *Local Input*

Note: D **must not be changed** between calls to the factorize and the solve routines.

Note: the dimension of array D must be at least N_b^A.

On entry: the local part of the distributed vector d which contains the information about the factorization of the matrix A_s as returned by F07JRFP (PZPTTRF).

5: E($*$) — COMPLEX*16 array *Local Input*

Note: E **must not be changed** between calls to the factorize and the solve routines.

Note: the dimension of array E must be at least N_b^A.

On entry: the local part of the distributed vector e which contains the information about the factorization of the matrix A_s as returned by F07JRFP (PZPTTRF).

6: JA — INTEGER *Global Input*

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

Constraints: $1 \leq$ JA $\leq n_A -$ N+1.

7: IDESCA(∗) — INTEGER array *Local Input*

Note: the dimension of the array IDESCA must be at least 5 when IDESCA(1) = 501 or 502 and must be at least 9 when IDESCA(1) = 1.

Distribution: if IDESCA(1) = 1, the array elements IDESCA(3:8) must be global to the processor grid. If IDESCA(1) = 501 or 502, then only the array elements IDESCA(3:5) must be global. In either case IDESCA(2) is 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.

If IDESCA(1) = 1, then $p = 1 \times n_p$ and:

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 (in that case, IDESCA(5) = 1);

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 (since the logical grid is one-dimensional, IDESCA(7) = 0);

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 (in this case, IDESCA(9) is not referenced).

If IDESCA(1) = 501 or 502, then $p = 1 \times n_p$ or $p = m_p \times 1$, and:

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

IDESCA(3), the size n_A, of the matrix A;

IDESCA(4), the blocking factor, N_b^A, used to distribute the matrix A;

IDESCA(5), the processor column index over which the first element of the matrix A is distributed;

IDESCA(6:9) are not referenced.

Suggested value: IDESCA(1) = 501 and $p = 1 \times n_p$.

Constraints:

IDESCA(1) = 1, 501 or 502;

if IDESCA(1) = 1, then $p = 1 \times n_p$;

if IDESCA(1) = 501 or 502; then $p = m_p \times 1$ or $p = 1 \times n_p$;

if IDESCA(1) = 501 or 502, then

$1 \leq$ IDESCA(3) $\leq$ N + JA−1;

IDESCA(4) ≥ 2 and $p\times$IDESCA(4) $\geq$ mod(JA−1,IDESCA(4)) + N;

IDESCA(5) ≥ 0;

if IDESCA(1) = 1, then

$1 \leq$ IDESCA(4) $\leq$ N + JA−1;

IDESCA(6) ≥ 2 and $p\times$IDESCA(6) $\geq$ mod(JA−1,IDESCA(4)) + N;

IDESCA(8) ≥ 0.

8: B(∗) — COMPLEX*16 array *Local Input/Local Output*

Note: the array B is formally defined as a vector. However, you may find it more convenient to consider B as a two-dimensional array of dimension (LDB,γ) where LDB = IDESCB(9) if IDESCB(1) = 1, or LDB = IDESCB(6) if IDESCB(1) = 502; and $\gamma \geq r$.

On entry: the local part of the right-hand side B which is stored in row block fashion.

On exit: the n by r solution matrix X distributed in the same row block distribution.

9: IB — INTEGER *Global Input*

On entry: i_B, the row index of the matrix B, that identifies the first row of the submatrix B_s.

Constraint: IB = JA.

10: IDESCB(∗) — INTEGER array *Local Input*

Note: the dimension of the array IDESCB must be at least 6 when IDESCB(1) = 502 and must be at least 9 when IDESCB(1) = 1.

Distribution: if IDESCB(1) = 1, the array elements IDESCB(3:8) must be global to the processor grid. If IDESCB(1) = 502, then only the array elements IDESCB(3:5) must be global. In either case IDESCB(2) is local to each processor.

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

IDESCB(1), the descriptor type.

If IDESCB(1) = 1, then $p = m_p \times 1$ and:

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

IDESCB(3), the number of rows, m_B, of the matrix B;

IDESCB(4), the number of columns, n_B, of the matrix B;

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

IDESCB(6), the blocking factor, N_b^B, used to distribute the columns of the matrix B (in that case, IDESCB(6) = 1);

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

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

IDESCB(9), the leading dimension of the conceptual two-dimensional array B.

If IDESCB(1) = 502, then $p = 1 \times n_p$, or $p = m_p \times 1$, and:

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

IDESCB(3), the size m_B, of the matrix B;

IDESCB(4), the blocking factor, M_b^B, used to distribute the matrix B;

IDESCB(5), the processor column index over which the first element of the matrix B is distributed;

IDESCB(6), the leading dimension of the conceptual two-dimensional array B;

IDESCB(7:9) are not referenced.

Suggested value: IDESCB(1) = 502 and $p = 1 \times n_p$.

Constraints:

IDESCB(1) = 1 or 502;

if IDESCB(1) = 1, or 502, then $p = 1 \times n_p$;

if IDESCB(1) = 502; then $p = m_p \times 1$;

if IDESCB(1) = 502, then

$1 \leq$ IDESCB(3) $\leq$ N + IB−1;

IDESCB(4) ≥ 2 and $p\times$IDESCB(4) $\geq$ mod(IB−1,IDESCB(4)) + N;

IDESCB(5) ≥ 0;

IDESCB(2) = IDESCA(2);

and if IDESCA(1) = 501 or 502, then

IDESCB(4) = IDESCA(4);

IDESCB(5) = IDESCA(5);

and if IDESCA(1) = 1, then

IDESCB(4) = IDESCA(6);
IDESCB(5) = IDESCA(8);
if IDESCB(1) = 1, then
$1 \le$ IDESCB(4) $\le$ N + IB -1;
IDESCB(6) ≥ 2 and $p\times$IDESCB(6) $\ge$ mod(IB$-$1,IDESCB(4)) + N;
IDESCB(8) ≥ 0;
IDESCB(2) = IDESCA(2);
and if IDESCA(1) = 1, then
IDESCB(6)= IDESCA(6);
IDESCB(8) = IDESCA(8);
and if IDESCA(1) = 501 or 502, then
IDESCB(6) = IDESCA(4);
IDESCB(8) = IDESCA(5).

11: AF($*$) — COMPLEX*16 array *Local Input*

Note: AF **must not be changed** between calls to the factorize and the solve routines.

On entry: the auxiliary fill-in space, created and stored by F07JRFP (PZPTTRF). If LAF is not large enough, after an unsuccesful exit, INT(AF(1)) will contain the minimum acceptable size of AF.

12: LAF — INTEGER *Local Input*

On entry: The dimension of the array AF .

Constraints: LAF $\ge 12 \times p + 3 \times N_b^A$.

13: WORK($*$) — COMPLEX*16 array *Local Workspace*

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 the real part of array element WORK(1).

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

14: 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 = (10+2$\times$min(100,NRHS))$\times$NPCOL + 4$\times$NRHS, where NRHS = n_r and NPCOL = n_p.

15: 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 $= -i$

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 occurred 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 total number of floating-point operations is approximately $16nr$.

6.1 Algorithmic Detail

Forward and backward substitution are used. Assuming the decomposition of the matrix $A_s = PU^HUP^T = PLL^HP^T$, where U is upper triangular, L is lower triangular, P is a permutation matrix, and $L = U^H$ (because the matrix A_s is tridiagonal and Hermitian); the solution X is computed by solving $PU^HY = B$ and then $UP^TX = Y$.

6.2 Parallelism Detail

The Level-3 BLAS operations are carried out 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 $(A + \Sigma)x = b$, where

$$|\Sigma| \leq c(n)\epsilon|U^T| \cdot |U|;$$

$c(n)$ is a modest linear function of n and ϵ is the ***machine precision***. If x is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x - \hat{x}\|_\infty}{\|x\|_\infty} \leq \epsilon c(n)\kappa(A),$$

where $\kappa(A)$ is the condition number of A. See the F07 Chapter Introduction.

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] 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

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

8 Example

See Section 8 of the document for F07JRFP (PZPTTRF).