F08JJFP (PDSTEBZ)

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

F08JJFP (PDSTEBZ) computes eigenvalues of a real symmetric tridiagonal matrix T in parallel by bisection. There are options to request all eigenvalues, all eigenvalues in the interval $(v_l, v_u]$, or the eigenvalues indexed i_l through i_u.

Each logical processor must contain the vectors d and e which are the diagonal and off-diagonal elements of T, respectively. A static partitioning of the workload is performed at the beginning of this routine which results in all processes finding an (approximately) equal number of eigenvalues.

2 Specification

```
      SUBROUTINE F08JJFP(ICNTXT, RANGE, ORDER, N, VL, VU, IL, IU,
     1                   ABSTOL, D, E, M, NSPLIT, W, IBLOCK, ISPLIT,
     2                   WORK, LWORK, IWORK, LIWORK, INFO)
      ENTRY      PDSTEBZ(ICNTXT, RANGE, ORDER, N, VL, VU, IL, IU,
     1                   ABSTOL, D, E, M, NSPLIT, W, IBLOCK, ISPLIT,
     2                   WORK, LWORK, IWORK, LIWORK, INFO)
      DOUBLE PRECISION   VL, VU, ABSTOL, D(*), E(*), W(*), WORK(LWORK)
      INTEGER            ICNTXT, N, IL, IU, M, NSPLIT, IBLOCK(*),
     1                   ISPLIT(*), LWORK, IWORK(LIWORK), LIWORK, INFO
      CHARACTER*1        RANGE, ORDER
```

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

3 Data Distribution

3.1 Definitions

None.

3.2 Global and Local Arguments

The input arguments RANGE, ORDER, N, VL, VU, IL, IU, ABSTOL, D and E are global and so must have the same value on entry to the routine on each processor. The output arguments M, NSPLIT, W, IBLOCK, ISPLIT and INFO are also global and so will have the same value on exit from the routine on each processor. The remaining arguments are local.

3.3 Distribution Strategy

Each logical processor (defined by the BLACS context ICNTXT) must contain the vectors d and e which are the diagonal and off-diagonal entries of T, respectively.

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: RANGE — CHARACTER*1 *Global Input*

On entry: indicates which eigenvalues are required as follows:

if RANGE = 'A', then all the eigenvalues are required;
if RANGE = 'V', then all the eigenvalues in the half-open interval $(v_l, v_u]$ are required;
if RANGE = 'I', then eigenvalues with indices i_l to i_u are required.

Constraint: RANGE = 'A', 'V' or 'I'.

3: ORDER — CHARACTER*1 *Global Input*

On entry: indicates the order in which the eigenvalues and their block numbers are to be stored as follows:

if ORDER = 'B', then the eigenvalues are to be grouped by split-off block and ordered from smallest to largest within each block;
if ORDER = 'E', then the eigenvalues for the entire matrix are to be ordered from smallest to largest.

Constraint: ORDER = 'B' or 'E'.

4: N — INTEGER *Global Input*

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

Constraint: N ≥ 0.

5: VL — DOUBLE PRECISION *Global Input*
6: VU — DOUBLE PRECISION *Global Input*

On entry: if RANGE = 'V', the lower and the upper bounds, respectively, of the half-open interval $(v_l, v_u]$ within which the required eigenvalues lie.

Not referenced if RANGE = 'A' or 'I'.

Constraint: VL < VU if RANGE = 'V'.

7: IL — INTEGER *Global Input*
8: IU — INTEGER *Global Input*

On entry: if RANGE = 'I', u_l and i_u, the indices of the first and the last eigenvalues, respectively, to be computed (assuming that the eigenvalues are in non-decreasing order).

Not referenced if RANGE = 'A' or 'V'.

Constraint: $1 \leq$ IL $\leq$ IU $\leq$ N if RANGE = 'I'.

9: ABSTOL — DOUBLE PRECISION *Global Input*

On entry: the absolute tolerance to which each eigenvalue is required. An eigenvalue (or cluster) is considered to have converged if it lies in an interval of width $\leq$ ABSTOL. If ABSTOL ≤ 0.0, then the tolerance is taken as ***machine precision*** $\times\|T\|_1$.

10: D($*$) — DOUBLE PRECISION array *Global Input*

Note: the dimension of the array D must be at least max(1,N).

On entry: d, the diagonal elements of the tridiagonal matrix T.

11: E($*$) — DOUBLE PRECISION array *Global Input*

Note: the dimension of the array E must be at least max(1,N − 1).

On entry: e, the off-diagonal elements of the tridiagonal matrix T.

12: M — INTEGER *Global Output*

On exit: the actual number of eigenvalues found.

13: NSPLIT — INTEGER *Global Output*

On exit: the number of diagonal blocks which constitute the tridiagonal matrix T.

14: W(∗) — DOUBLE PRECISION array *Global Output*

Note: the dimension of the array W must be at least max(1,N).

On exit: the required eigenvalues of the tridiagonal matrix T stored in W(1) to W(M).

15: IBLOCK(∗) — INTEGER array *Global Output*

Note: the dimension of the array IBLOCK must be at least max(1,N).

On exit: at each row/column j where $e(j)$ is zero or negligible, the matrix T is considered to split into a block diagonal matrix and IBLOCK(i) contains the block number of the eigenvalue stored in W(i), for $i = 1, 2, \ldots, m$. Note that IBLOCK(i) < 0 for some i whenever INFO = 1 (see Section 5).

16: ISPLIT(∗) — INTEGER array *Global Output*

Note: the dimension of the array ISPLIT must be at least max(1,N).

On exit: the leading NSPLIT elements contain the points at which the matrix T splits up into submatrices (blocks) as follows. The first submatrix consists of rows/columns 1 to ISPLIT(1), the second submatrix consists of rows/columns ISPLIT(1) + 1 to ISPLIT(2), ..., and the last submatrix consists of rows/columns ISPLIT(NSPLIT − 1) + 1 to ISPLIT(NSPLIT) (= n).

17: WORK(LWORK) — DOUBLE PRECISION array *Local Workspace*
18: LWORK — INTEGER *Local Input*

On entry: the size of the array WORK.

Constraint: LWORK ≥ max(7,5×N).

19: IWORK(LIWORK) — INTEGER array *Local Workspace*
20: LIWORK — INTEGER *Local Input*

On entry: the size of the array IWORK.

Constraint: LIWORK ≥ max(14,4×N).

21: INFO — INTEGER *Global Output*

On exit: INFO = 0 unless the routine detects an error (see Section 5).

5 Errors and Warnings

If INFO ≠ 0 an explanatory message is output and control returned to the calling program.

INFO = $-i$

On entry, the ith argument was invalid. 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. An explanatory message distinguishes between these two cases.

INFO = 1

The algorithm failed to compute some (or all) of the required eigenvalues to the desired accuracy. More precisely, IBLOCK(i) < 0 indicates that the ith eigenvalue (stored in W(i)) failed to converge. The effect is that the eigenvalues may not be as accurate as specified by the absolute and relative tolerances.

INFO = 2

There is a mismatch between the number of eigenvalues computed and the desired number.

INFO = 3

No eigenvalues have been computed. The floating-point arithmetic on the computer is not behaving as expected.

If failures with INFO ≥ 1 are causing persistent trouble and the user has checked that the routine is being called correctly, please contact NAG.

6 Further Comments

6.1 Algorithmic Detail

The diagonal matrix $D(\tau)$ of the LDL^T of the shifted tridiagonal matrix $T - \tau I$ are computed where τ is a shift. The inertia of the matrix $T - \tau I$ is then given by $D(\tau)$. The number of negative values of the diagonal entries of $D(\tau)$ gives the number of eigenvalues of T in the interval $(-\infty, \tau]$ which is the crucial information required to locate eigenvalues. See Kahan [1].

6.2 Parallelism Detail

The algorithm is embarrassingly parallel and the inter-process communication is minimal.

6.3 Accuracy

The remarks assume that ABSTOL is set to the underflow threshold which can be obtained by calling the function X02AKF. The eigenvalue with the largest magnitude of the tridiagonal form T can be computed to high relative accuracy. Typically, this eigenvalue is accurate to 4 ulps (units in the last place held). The eigenvalues with smaller magnitudes will suffer absolute errors which are no larger than the error in the largest eigenvalue. However, the reduction to the tridiagonal form (prior to the calling routine) may exclude the possibility of obtaining high relative accuracy for the eigenvalues of the original matrix.

7 References

[1] Kahan W (1966) Accurate eigenvalues of a symmetric tridiagonal matrix *Report CS41* Stanford University

8 Example

The example program illustrates the computation of eigenvalues of a 10 by 10 dense symmetric matrix A_s. The (i, j)th element of this matrix A_s is taken to be $\max(i, j)$ and the matrix is generated using the routine F01ZQFP.

The matrix A_s is first tridiagonalised using the routine F08FEFP (PDSYTRD).

The diagonal vector d and the off-diagonal vector e are gathered to each and every logical processor by calling the routine F01ZPFP twice.

Finally, the vector d (denoted locally by the array DL) and the the vector e (denoted locally by the array EL) are used by the routine F08JJFP (PDSTEBZ) to compute the eigenvalues of the tridiagonal T.

8.1 Example Text

```
*     F08JJFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
      INTEGER          N
      PARAMETER        (N=10)
      INTEGER          NA
      PARAMETER        (NA=20)
      INTEGER          MG, NG
      PARAMETER        (MG=2,NG=2)
      INTEGER          NB
      PARAMETER        (NB=2)
      INTEGER          LDA, TDA
      PARAMETER        (LDA=NA/MG+NB,TDA=NA/NG+NB)
      INTEGER          LWORK, LIWORK
      PARAMETER        (LWORK=50,LIWORK=50)
```

```
*     .. Local Scalars ..
      INTEGER          I, IA, ICNTXT, IFAIL, INFO, JA, M, NCOLS, NROWS,
     +                 NSPLIT
      LOGICAL          ROOT
      CHARACTER        UPLO
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,TDA), D(TDA), DL(N), E(TDA), EL(N),
     +                 TAU(TDA), W(N), WORK(LWORK)
      INTEGER          IBLOCK(N), IDESCA(9), ISPLIT(N), IWORK(LIWORK)
*     .. External Functions ..
      LOGICAL          Z01ACFP
      EXTERNAL         Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         F01ZPFP, F01ZQFP, F08FEFP, F08JJFP, GMATA,
     +                 Z01AAFP, Z01ABFP
*     .. Executable Statements ..
*
      ROOT = Z01ACFP()
      IF (ROOT) THEN
         WRITE (NOUT,*) 'F08JJFP Example Program Results'
         WRITE (NOUT,*)
      END IF
*
      NROWS = MG
      NCOLS = NG
      IFAIL = 0
      CALL Z01AAFP(ICNTXT,NROWS,NCOLS,IFAIL)
*
      IA = 1
      JA = 1
      IDESCA(1) = 1
      IDESCA(2) = ICNTXT
      IDESCA(3) = NA
      IDESCA(4) = NA
      IDESCA(5) = NB
      IDESCA(6) = NB
      IDESCA(7) = 0
      IDESCA(8) = 0
      IDESCA(9) = LDA

*
*     Generate the matrix A
*
      IFAIL = 0
      CALL F01ZQFP(GMATA,N,N,A,IA,JA,IDESCA,IFAIL)
*
*        Reduce to the tridiagonal form
*
      UPLO = 'U'
*
      CALL F08FEFP(UPLO,N,A,IA,JA,IDESCA,D,E,TAU,WORK,LWORK,INFO)
*
*        Gather the diagonal D to each logical processor
*
      IFAIL = 0
      CALL F01ZPFP(N,IA,JA,IDESCA,D,DL,WORK,LWORK,IFAIL)
*
*        Gather the off-diagonal E to each logical processor
```

```
*
      IFAIL = 0
      CALL F01ZPFP(N,IA,JA,IDESCA,E,EL,WORK,LWORK,IFAIL)
*
*     Compute eigenvalues
*
      IF (UPLO.EQ.'U') THEN
         CALL F08JJFP(ICNTXT,'A','E',N,0.0D0,0.0D0,0,0,-1.0D0,DL,EL,M,
     +                NSPLIT,W,IBLOCK,ISPLIT,WORK,LWORK,IWORK,LIWORK,
     +                INFO)
      ELSE IF (UPLO.EQ.'L') THEN
         CALL F08JJFP(ICNTXT,'A','E',N,0.0D0,0.0D0,0,0,-1.0D0,DL,EL(2),
     +                M,NSPLIT,W,IBLOCK,ISPLIT,WORK,LWORK,IWORK,LIWORK,
     +                INFO)
      END IF
*
*     Print the computed eigenvalues
*
      IF (ROOT) THEN
         WRITE (NOUT,*) 'Eigenvalues'
         WRITE (NOUT,*)
         DO 20 I = 1, N
            WRITE (NOUT,'(1X,I3,5X,F12.4)') I, W(I)
   20    CONTINUE
      END IF
      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'N',IFAIL)
*
      STOP
      END
*
      SUBROUTINE GMATA(I1,I2,J1,J2,AL,LDAL)
*     GMATA generates the block A(I1: I2, J1: J2) of the matrix A such
*     that
*
*        a(i,j) = max(i,j)
*
*     in the array AL.
*
*     .. Scalar Arguments ..
      INTEGER          I1, I2, J1, J2, LDAL
*     .. Array Arguments ..
      DOUBLE PRECISION AL(LDAL,*)
*     .. Local Scalars ..
      INTEGER          I, J, K, L
*     .. Intrinsic Functions ..
      INTRINSIC        MAX
*     .. Executable Statements ..
      L = 1
      DO 40 J = J1, J2
         K = 1
         DO 20 I = I1, I2
            AL(K,L) = MAX(I,J)
            K = K + 1
   20    CONTINUE
         L = L + 1
   40 CONTINUE
*
```

```
      RETURN
*
      END
```

8.2 Example Data

None.

8.3 Example Results

```
F08JJFP Example Program Results

Eigenvalues

   1         -7.9471
   2         -2.2404
   3         -1.0915
   4         -0.6715
   5         -0.4749
   6         -0.3702
   7         -0.3108
   8         -0.2742
   9          7.2024
  10         61.1782
```