NAG Library Routine Document

F07CPF (ZGTSVX)

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F07CPF (ZGTSVX) uses the LU factorization to compute the solution to a complex system of linear equations

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

where A is a tridiagonal matrix of order n and X and B are n by r matrices. Error bounds on the solution and a condition estimate are also provided.

2 Specification

```
SUBROUTINE F07CPF (FACT, TRANS, N, NRHS, DL, D, DU, DLF, DF, DUF, DU2,         &
                   IPIV, B, LDB, X, LDX, RCOND, FERR, BERR, WORK, RWORK,        &
                   INFO)

INTEGER                N, NRHS, IPIV(*), LDB, LDX, INFO
REAL (KIND=nag_wp)     RCOND, FERR(NRHS), BERR(NRHS), RWORK(N)
COMPLEX (KIND=nag_wp)  DL(*), D(*), DU(*), DLF(*), DF(*), DUF(*), DU2(*),      &
                       B(LDB,*), X(LDX,*), WORK(2*N)
CHARACTER(1)           FACT, TRANS
```

The routine may be called by its LAPACK name ***zgtsvx***.

3 Description

F07CPF (ZGTSVX) performs the following steps:

1. If FACT = 'N', the LU decomposition is used to factor the matrix A as $A = LU$, where L is a product of permutation and unit lower bidiagonal matrices and U is upper triangular with nonzeros in only the main diagonal and first two superdiagonals.
2. If some $u_{ii} = 0$, so that U is exactly singular, then the routine returns with INFO $= i$. Otherwise, the factored form of A is used to estimate the condition number of the matrix A. If the reciprocal of the condition number is less than ***machine precision***, INFO $= \mathrm{N} + 1$ is returned as a warning, but the routine still goes on to solve for X and compute error bounds as described below.
3. The system of equations is solved for X using the factored form of A.
4. Iterative refinement is applied to improve the computed solution matrix and to calculate error bounds and backward error estimates for it.

4 References

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 http://www.netlib.org/lapack/lug

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

Higham N J (2002) *Accuracy and Stability of Numerical Algorithms* (2nd Edition) SIAM, Philadelphia

5 Parameters

1: FACT – CHARACTER(1) *Input*

On entry: specifies whether or not the factorized form of the matrix A has been supplied.

FACT = 'F'
: DLF, DF, DUF, DU2 and IPIV contain the factorized form of the matrix A. DLF, DF, DUF, DU2 and IPIV will not be modified.

FACT = 'N'
: The matrix A will be copied to DLF, DF and DUF and factorized.

Constraint: FACT = 'F' or 'N'.

2: TRANS – CHARACTER(1) *Input*

On entry: specifies the form of the system of equations.

TRANS = 'N'
: $AX = B$ (No transpose).

TRANS = 'T'
: $A^{\mathrm{T}}X = B$ (Transpose).

TRANS = 'C'
: $A^{\mathrm{H}}X = B$ (Conjugate transpose).

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

3: N – INTEGER *Input*

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

Constraint: $\mathrm{N} \geq 0$.

4: NRHS – INTEGER *Input*

On entry: r, the number of right-hand sides, i.e., the number of columns of the matrix B.

Constraint: $\mathrm{NRHS} \geq 0$.

5: DL($*$) – COMPLEX (KIND=nag_wp) array *Input*

Note: the dimension of the array DL must be at least $\max(1, \mathrm{N} - 1)$.

On entry: the $(n - 1)$ subdiagonal elements of A.

6: D($*$) – COMPLEX (KIND=nag_wp) array *Input*

Note: the dimension of the array D must be at least $\max(1, \mathrm{N})$.

On entry: the n diagonal elements of A.

7: DU($*$) – COMPLEX (KIND=nag_wp) array *Input*

Note: the dimension of the array DU must be at least $\max(1, \mathrm{N} - 1)$.

On entry: the $(n - 1)$ superdiagonal elements of A.

8: DLF($*$) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array DLF must be at least $\max(1, \mathrm{N} - 1)$.

On entry: if FACT = 'F', DLF contains the $(n - 1)$ multipliers that define the matrix L from the LU factorization of A.

On exit: if FACT = 'N', DLF contains the $(n - 1)$ multipliers that define the matrix L from the LU factorization of A.

9: DF(*) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'F', DF contains the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

On exit: if FACT = 'N', DF contains the n diagonal elements of the upper triangular matrix U from the LU factorization of A.

10: DUF(*) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On entry: if FACT = 'F', DUF contains the $(n-1)$ elements of the first superdiagonal of U.

On exit: if FACT = 'N', DUF contains the $(n-1)$ elements of the first superdiagonal of U.

11: DU2(*) – COMPLEX (KIND=nag_wp) array *Input/Output*

Note: the dimension of the array DU2 must be at least max(1, N − 2).

On entry: if FACT = 'F', DU2 contains the $(n-2)$ elements of the second superdiagonal of U.

On exit: if FACT = 'N', DU2 contains the $(n-2)$ elements of the second superdiagonal of U.

12: IPIV(*) – INTEGER array *Input/Output*

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

On entry: if FACT = 'F', IPIV contains the pivot indices from the LU factorization of A.

On exit: if FACT = 'N', IPIV contains the pivot indices from the LU factorization of A; row i of the matrix was interchanged with row IPIV(i). IPIV(i) will always be either i or $i+1$; IPIV(i) = i indicates a row interchange was not required.

13: B(LDB,*) – COMPLEX (KIND=nag_wp) array *Input*

Note: the second dimension of the array B must be at least max(1, NRHS).

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

14: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07CPF (ZGTSVX) is called.

Constraint: LDB ≥ max(1, N).

15: X(LDX,*) – COMPLEX (KIND=nag_wp) array *Output*

Note: the second dimension of the array X must be at least max(1, NRHS).

On exit: if INFO = 0 or N + 1, the n by r solution matrix X.

16: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which F07CPF (ZGTSVX) is called.

Constraint: LDX ≥ max(1, N).

17: RCOND – REAL (KIND=nag_wp) *Output*

On exit: the estimate of the reciprocal condition number of the matrix A. If RCOND = 0.0, the matrix may be exactly singular. This condition is indicated by INFO > 0 and INFO ≤ N. Otherwise, if RCOND is less than the ***machine precision***, the matrix is singular to working precision. This condition is indicated by INFO = N + 1.

18: FERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or $\mathrm{N}+1$, an estimate of the forward error bound for each computed solution vector, such that $\|\hat{x}_j - x_j\|_\infty / \|x_j\|_\infty \leq \mathrm{FERR}(j)$ where $\hat{x}_j$ is the jth column of the computed solution returned in the array X and x_j is the corresponding column of the exact solution X. The estimate is as reliable as the estimate for RCOND, and is almost always a slight overestimate of the true error.

19: BERR(NRHS) – REAL (KIND=nag_wp) array *Output*

On exit: if INFO $= 0$ or $\mathrm{N}+1$, an estimate of the component-wise relative backward error of each computed solution vector $\hat{x}_j$ (i.e., the smallest relative change in any element of A or B that makes $\hat{x}_j$ an exact solution).

20: WORK($2 \times \mathrm{N}$) – COMPLEX (KIND=nag_wp) array *Workspace*

21: RWORK(N) – REAL (KIND=nag_wp) array *Workspace*

22: INFO – INTEGER *Output*

On exit: INFO $= 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFO < 0

If INFO $= -i$, the ith argument had an illegal value. An explanatory message is output, and execution of the program is terminated.

INFO > 0 and INFO $\leq$ N

If INFO $= i$, $u(i,i)$ is exactly zero. The factorization has not been completed unless $i = \mathrm{N}$, but the factor U is exactly singular, so the solution and error bounds could not be computed. RCOND $= 0.0$ is returned.

INFO $= \mathrm{N}+1$

The triangular matrix U is nonsingular, but RCOND is less than ***machine precision***, meaning that the matrix is singular to working precision. Nevertheless, the solution and error bounds are computed because there are a number of situations where the computed solution can be more accurate than the value of RCOND would suggest.

7 Accuracy

For each right-hand side vector b, the computed solution $\hat{x}$ is the exact solution of a perturbed system of equations $(A+E)\hat{x} = b$, where

$$|E| \leq c(n)\epsilon|L||U|,$$

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***. See Section 9.3 of Higham (2002) for further details.

If x is the true solution, then the computed solution $\hat{x}$ satisfies a forward error bound of the form

$$\frac{\|x-\hat{x}\|_\infty}{\|\hat{x}\|_\infty} \leq w_c \operatorname{cond}(A,\hat{x},b)$$

where $\operatorname{cond}(A,\hat{x},b) = \left\| |A^{-1}|(|A||\hat{x}| + |b|) \right\|_\infty / \|\hat{x}\|_\infty \leq \operatorname{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$. If $\hat{x}$ is the jth column of X, then w_c is returned in BERR(j) and a bound on $\|x-\hat{x}\|_\infty / \|\hat{x}\|_\infty$ is returned in FERR(j). See Section 4.4 of Anderson *et al.* (1999) for further details.

8 Further Comments

The total number of floating point operations required to solve the equations $AX = B$ is proportional to nr.

The condition number estimation typically requires between four and five solves and never more than eleven solves, following the factorization. The solution is then refined, and the errors estimated, using iterative refinement.

In practice the condition number estimator is very reliable, but it can underestimate the true condition number; see Section 15.3 of Higham (2002) for further details.

The real analogue of this routine is F07CBF (DGTSVX).

9 Example

This example solves the equations

$$AX = B,$$

where A is the tridiagonal matrix

$$A = \begin{pmatrix} -1.3+1.3i & 2.0-1.0i & 0 & 0 & 0 \\ 1.0-2.0i & -1.3+1.3i & 2.0+1.0i & 0 & 0 \\ 0 & 1.0+1.0i & -1.3+3.3i & -1.0+1.0i & 0 \\ 0 & 0 & 2.0-3.0i & -0.3+4.3i & 1.0-1.0i \\ 0 & 0 & 0 & 1.0+1.0i & -3.3+1.3i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 2.4-5.0i & 2.7+6.9i \\ 3.4+18.2i & -6.9-5.3i \\ -14.7+9.7i & -6.0-0.6i \\ 31.9-7.7i & -3.9+9.3i \\ -1.0+1.6i & -3.0+12.2i \end{pmatrix}.$$

Estimates for the backward errors, forward errors and condition number are also output.

9.1 Program Text

```
    Program f07cpfe

!     F07CPF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zgtsvx
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rcond
      Integer                          :: i, ifail, info, ldb, ldx, n, nrhs
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: b(:,:), d(:), df(:), dl(:),       &
                                          dlf(:), du(:), du2(:), duf(:),     &
                                          work(:), x(:,:)
      Real (Kind=nag_wp), Allocatable  :: berr(:), ferr(:), rwork(:)
      Integer, Allocatable             :: ipiv(:)
      Character (1)                    :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07CPF Example Program Results'
      Write (nout,*)
      Flush (nout)
!     Skip heading in data file
```

```
      Read (nin,*)
      Read (nin,*) n, nrhs
      ldb = n
      ldx = n
      Allocate (b(ldb,nrhs),d(n),df(n),dl(n-1),dlf(n-1),du(n-1),du2(n-2), &
        duf(n-1),work(2*n),x(ldx,nrhs),berr(nrhs),ferr(nrhs),rwork(n),ipiv(n))

!     Read the tridiagonal matrix A from data file

      Read (nin,*) du(1:n-1)
      Read (nin,*) d(1:n)
      Read (nin,*) dl(1:n-1)

!     Read the right hand matrix B

      Read (nin,*)(b(i,1:nrhs),i=1,n)

!     Solve the equations AX = B
!     The NAG name equivalent of zgtsvx is f07cpf
      Call zgtsvx('No factors','No transpose',n,nrhs,dl,d,du,dlf,df,duf,du2, &
        ipiv,b,ldb,x,ldx,rcond,ferr,berr,work,rwork,info)

      If ((info==0) .Or. (info==n+1)) Then

!       Print solution, error bounds and condition number

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call x04dbf('General',' ',n,nrhs,x,ldx,'Bracketed','F7.4', &
          'Solution(s)','Integer',rlabs,'Integer',clabs,80,0,ifail)

        Write (nout,*)
        Write (nout,*) 'Backward errors (machine-dependent)'
        Write (nout,99999) berr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimated forward error bounds (machine-dependent)'
        Write (nout,99999) ferr(1:nrhs)
        Write (nout,*)
        Write (nout,*) 'Estimate of reciprocal condition number'
        Write (nout,99999) rcond

        If (info==n+1) Then
          Write (nout,*)
          Write (nout,*) 'The matrix A is singular to working precision'
        End If
      Else
        Write (nout,99998) 'The (', info, ',', info, ')', &
          ' element of the factor U is zero'
      End If

99999 Format ((3X,1P,7E11.1))
99998 Format (1X,A,I3,A,I3,A,A)
    End Program f07cpfe
```

9.2 Program Data

```
F07CPF Example Program Data
   5    2                                                 :Values of N and NRHS
 (  2.0, -1.0) (  2.0,  1.0) ( -1.0,  1.0) (  1.0, -1.0) :End of DU
 ( -1.3,  1.3) ( -1.3,  1.3) ( -1.3,  3.3) ( -0.3,  4.3)
 ( -3.3,  1.3)                                           :End of D
 (  1.0, -2.0) (  1.0 , 1.0) (  2.0, -3.0) (  1.0,  1.0) :End of  DL
 (  2.4, -5.0) (  2.7,  6.9)
 (  3.4, 18.2) ( -6.9, -5.3)
 (-14.7,  9.7) ( -6.0, -0.6)
 ( 31.9, -7.7) ( -3.9,  9.3)
 ( -1.0,  1.6) ( -3.0, 12.2)                             :End of B
```

9.3 Program Results

```
 F07CPF Example Program Results

 Solution(s)
                    1                  2
 1  ( 1.0000, 1.0000) ( 2.0000,-1.0000)
 2  ( 3.0000,-1.0000) ( 1.0000, 2.0000)
 3  ( 4.0000, 5.0000) (-1.0000, 1.0000)
 4  (-1.0000,-2.0000) ( 2.0000, 1.0000)
 5  ( 1.0000,-1.0000) ( 2.0000,-2.0000)

 Backward errors (machine-dependent)
       3.7E-17    6.7E-17

 Estimated forward error bounds (machine-dependent)
       5.4E-14    7.3E-14

 Estimate of reciprocal condition number
       5.4E-03
```