

NAG Library Routine Document

F07CSF (ZGTTRS)

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

F07CSF (ZGTTRS) computes the solution to a complex system of linear equations $AX = B$ or $A^T X = B$ or $A^H X = B$, where A is an n by n tridiagonal matrix and X and B are n by r matrices, using the LU factorization returned by F07CRF (ZGTTRF).

2 Specification

```
SUBROUTINE F07CSF (TRANS, N, NRHS, DL, D, DU, DU2, IPIV, B, LDB, INFO)
INTEGER          N, NRHS, IPIV(*), LDB, INFO
COMPLEX (KIND=nag_wp) DL(*), D(*), DU(*), DU2(*), B(LDB,*)
CHARACTER(1)    TRANS
```

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

3 Description

F07CSF (ZGTTRS) should be preceded by a call to F07CRF (ZGTTRF), which uses Gaussian elimination with partial pivoting and row interchanges to factorize the matrix A as

$$A = PLU,$$

where P is a permutation matrix, L is unit lower triangular with at most one nonzero subdiagonal element in each column, and U is an upper triangular band matrix, with two superdiagonals. F07CSF (ZGTTRS) then utilizes the factorization to solve the required equations.

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

5 Parameters

1: TRANS – CHARACTER(1) *Input*

On entry: specifies the equations to be solved as follows:

TRANS = 'N'

Solve $AX = B$ for X .

TRANS = 'T'

Solve $A^T X = B$ for X .

TRANS = 'C'

Solve $A^H X = B$ for X .

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

- 2: N – INTEGER *Input*
On entry: n , the order of the matrix A .
Constraint: $N \geq 0$.
- 3: NRHS – INTEGER *Input*
On entry: r , the number of right-hand sides, i.e., the number of columns of the matrix B .
Constraint: $NRHS \geq 0$.
- 4: DL(*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the dimension of the array DL must be at least $\max(1, N - 1)$.
On entry: must contain the $(n - 1)$ multipliers that define the matrix L of the LU factorization of A .
- 5: D(*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the dimension of the array D must be at least $\max(1, N)$.
On entry: must contain the n diagonal elements of the upper triangular matrix U from the LU factorization of A .
- 6: DU(*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the dimension of the array DU must be at least $\max(1, N - 1)$.
On entry: must contain the $(n - 1)$ elements of the first superdiagonal of U .
- 7: DU2(*) – COMPLEX (KIND=nag_wp) array *Input*
Note: the dimension of the array DU2 must be at least $\max(1, N - 2)$.
On entry: must contain the $(n - 2)$ elements of the second superdiagonal of U .
- 8: IPIV(*) – INTEGER array *Input*
Note: the dimension of the array IPIV must be at least $\max(1, N)$.
On entry: must contain the n pivot indices that define the permutation matrix P . At the i th step, row i of the matrix was interchanged with row $IPIV(i)$, and $IPIV(i)$ must always be either i or $(i + 1)$, $IPIV(i) = i$ indicating that a row interchange was not performed.
- 9: B(LDB, *) – COMPLEX (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array B must be at least $\max(1, NRHS)$.
On entry: the n by r matrix of right-hand sides B .
On exit: the n by r solution matrix X .
- 10: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F07CSF (ZGTTRS) is called.
Constraint: $LDB \geq \max(1, N)$.
- 11: INFO – INTEGER *Output*
On exit: $INFO = 0$ unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

INFO < 0

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

7 Accuracy

The computed solution for a single right-hand side, \hat{x} , satisfies an equation of the form

$$(A + E)\hat{x} = b,$$

where

$$\|E\|_1 = O(\epsilon)\|A\|_1$$

and ϵ is the *machine precision*. An approximate error bound for the computed solution is given by

$$\frac{\|\hat{x} - x\|_1}{\|x\|_1} \leq \kappa(A) \frac{\|E\|_1}{\|A\|_1},$$

where $\kappa(A) = \|A^{-1}\|_1 \|A\|_1$, the condition number of A with respect to the solution of the linear equations. See Section 4.4 of Anderson *et al.* (1999) for further details.

Following the use of this routine F07CUF (ZGTCON) can be used to estimate the condition number of A and F07CVF (ZGTRFS) can be used to obtain approximate error bounds.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The total number of floating-point operations required to solve the equations $AX = B$ or $A^T X = B$ or $A^H X = B$ is proportional to nr .

The real analogue of this routine is F07CEF (DGTTRS).

10 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

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

10.1 Program Text

```

Program f07csfe

!      F07CSF Example Program Text

!      Mark 25 Release. NAG Copyright 2014.

!      .. Use Statements ..
Use nag_library, Only: nag_wp, x04dbf, zgttrf, zgttrs
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
Integer                    :: i, ifail, info, ldb, n, nrhs
!      .. Local Arrays ..
Complex (Kind=nag_wp), Allocatable :: b(:,,:), d(:), dl(:), du(:), du2(:)
Integer, Allocatable       :: ipiv(:)
Character (1)              :: clabs(1), rlabs(1)
!      .. Executable Statements ..
Write (nout,*) 'F07CSF Example Program Results'
Write (nout,*)
Flush (nout)
!      Skip heading in data file
Read (nin,*)
Read (nin,*) n, nrhs
ldb = n
Allocate (b(ldb,nrhs),d(n),dl(n-1),du(n-1),du2(n-2),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)

!      Factorize the tridiagonal matrix A
!      The NAG name equivalent of zgttrf is f07crf
Call zgttrf(n,dl,d,du,du2,ipiv,info)

If (info==0) Then

!      Solve the equations AX = B
!      The NAG name equivalent of zgttrs is f07csf
Call zgttrs('No transpose',n,nrhs,dl,d,du,du2,ipiv,b,ldb,info)

!      Print the solution

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

Else
  Write (nout,99999) 'The (', info, ', ', info, ')', &
    ' element of the factor U is zero'
End If

99999 Format (1X,A,I3,A,I3,A,A)
End Program f07csfe

```

10.2 Program Data

F07CSF 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

```

10.3 Program Results

F07CSF 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)

```
