NAG Library Routine Document

F07WSF (ZPFTRS)

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

F07WSF (ZPFTRS) solves a complex Hermitian positive definite system of linear equations with multiple right-hand sides,

$$AX = B,$$

using the Cholesky factorization computed by F07WRF (ZPFTRF) stored in Rectangular Full Packed (RFP) format.

2 Specification

```
SUBROUTINE F07WSF (TRANSR, UPLO, N, NRHS, AR, B, LDB, INFO)
INTEGER                N, NRHS, LDB, INFO
COMPLEX (KIND=nag_wp) AR(N*(N+1)/2), B(LDB,*)
CHARACTER(1)           TRANSR, UPLO
```

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

3 Description

F07WSF (ZPFTRS) is used to solve a complex Hermitian positive definite system of linear equations $AX = B$, the routine must be preceded by a call to F07WRF (ZPFTRF) which computes the Cholesky factorization of A, stored in RFP format. The RFP storage format is described in Section 3.3.3 in the F07 Chapter Introduction. The solution X is computed by forward and backward substitution.

If UPLO = 'U', $A = U^{\mathrm{H}}U$, where U is upper triangular; the solution X is computed by solving $U^{\mathrm{H}}Y = B$ and then $UX = Y$.

If UPLO = 'L', $A = LL^{\mathrm{H}}$, where L is lower triangular; the solution X is computed by solving $LY = B$ and then $L^{\mathrm{H}}X = Y$.

4 References

Gustavson F G, Waśniewski J, Dongarra J J and Langou J (2010) Rectangular full packed format for Cholesky's algorithm: factorization, solution, and inversion *ACM Trans. Math. Software* **37, 2**

5 Parameters

1: TRANSR – CHARACTER(1) *Input*

On entry: specifies whether the normal RFP representation of A or its conjugate transpose is stored.

TRANSR = 'N'
: The matrix A is stored in normal RFP format.

TRANSR = 'C'
: The conjugate transpose of the RFP representation of the matrix A is stored.

Constraint: TRANSR = 'N' or 'C'.

2: UPLO – CHARACTER(1) *Input*

On entry: specifies how A has been factorized.

UPLO = 'U'
: $A = U^{\mathrm{H}}U$, where U is upper triangular.

UPLO = 'L'
: $A = LL^{\mathrm{H}}$, where L is lower triangular.

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

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.

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

5: $\mathrm{AR}(\mathrm{N} \times (\mathrm{N}+1)/2)$ – COMPLEX (KIND=nag_wp) array *Input*

On entry: the Cholesky factorization of A stored in RFP format, as returned by F07WRF (ZPFTRF).

6: $\mathrm{B}(\mathrm{LDB}, *)$ – COMPLEX (KIND=nag_wp) array *Input/Output*

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

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

On exit: the n by r solution matrix X.

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F07WSF (ZPFTRS) is called.

Constraint: $\mathrm{LDB} \geq \max(1, \mathrm{N})$.

8: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

INFO < 0

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

7 Accuracy

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

if UPLO = 'U', $|E| \leq c(n)\epsilon|U^{\mathrm{H}}||U|$;

if UPLO = 'L', $|E| \leq c(n)\epsilon|L||L^{\mathrm{H}}|$,

$c(n)$ is a modest linear function of n, and ϵ is the ***machine precision***.

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

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

where $\mathrm{cond}(A,x) = \left\| |A^{-1}||A||x| \right\|_\infty / \|x\|_\infty \leq \mathrm{cond}(A) = \left\| |A^{-1}||A| \right\|_\infty \leq \kappa_\infty(A)$ and $\kappa_\infty(A)$ is the condition number when using the ∞-norm.

Note that $\mathrm{cond}(A,x)$ can be much smaller than $\mathrm{cond}(A)$.

8 Parallelism and Performance

F07WSF (ZPFTRS) is not threaded by NAG in any implementation.

F07WSF (ZPFTRS) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The total number of real floating-point operations is approximately $8n^2r$.

The real analogue of this routine is F07WEF (DPFTRS).

10 Example

This example solves the system of equations $AX = B$, where

$$A = \begin{pmatrix} 3.23 + 0.00i & 1.51 - 1.92i & 1.90 + 0.84i & 0.42 + 2.50i \\ 1.51 + 1.92i & 3.58 + 0.00i & -0.23 + 1.11i & -1.18 + 1.37i \\ 1.90 - 0.84i & -0.23 - 1.11i & 4.09 + 0.00i & 2.33 - 0.14i \\ 0.42 - 2.50i & -1.18 - 1.37i & 2.33 + 0.14i & 4.29 + 0.00i \end{pmatrix}$$

and

$$B = \begin{pmatrix} 3.93 - 6.14i & 1.48 + 6.58i \\ 6.17 + 9.42i & 4.65 - 4.75i \\ -7.17 - 21.83i & -4.91 + 2.29i \\ 1.99 - 14.38i & 7.64 - 10.79i \end{pmatrix}.$$

Here A is Hermitian positive definite, stored in RFP format, and must first be factorized by F07WRF (ZPFTRF).

10.1 Program Text

```
    Program f07wsfe

!     F07WSF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x04dbf, zpftrf, zpftrs
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter              :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                         :: i, ifail, info, k, lar1, ldb, lenar, &
                                         n, nrhs, q
      Character (1)                   :: transr, uplo
```

```
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: ar(:), b(:,:)
      Character (1)                      :: clabs(1), rlabs(1)
!     .. Executable Statements ..
      Write (nout,*) 'F07WSF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n, nrhs, uplo, transr

      lenar = n*(n+1)/2
      ldb = n
      Allocate (ar(lenar),b(ldb,nrhs))

!     Setup notional dimensions of RFP matrix AR
      k = n/2
      q = n - k
      If (transr=='N' .Or. transr=='n') Then
        lar1 = 2*k + 1
      Else
        lar1 = q
      End If

!     Read an RFP matrix into array AR
      Do i = 1, lar1
        Read (nin,*) ar(i:lenar:lar1)
      End Do

!     Read RHS matrix B
      Do i = 1, n
        Read (nin,*) b(i,1:nrhs)
      End Do

!     Factorize A
!     The NAG name equivalent of zpftrf is f07wrf
      Call zpftrf(transr,uplo,n,ar,info)

      Write (nout,*)
      Flush (nout)
      If (info==0) Then

!       Compute solution
!       The NAG name equivalent of zpftrs is f07wsf
        Call zpftrs(transr,uplo,n,nrhs,ar,b,ldb,info)

!       Print solution
        ifail = 0
        Call x04dbf('General',' ',n,nrhs,b,ldb,'Bracketed','F7.4', &
          'Solution(s)','Integer',rlabs,'Integer',clabs,80,0,ifail)

      Else
        Write (nout,*) 'A is not positive definite'
      End If

    End Program f07wsfe
```

10.2 Program Data

```
F07WSF Example Program Data
   4      2       'L'    'N'    : n, nrhs, uplo, transr
 ( 4.09,  0.00) ( 2.33, -0.14)
 ( 3.23,  0.00) ( 4.29,  0.00)
 ( 1.51,  1.92) ( 3.58,  0.00)
 ( 1.90, -0.84) (-0.23, -1.11)
 ( 0.42, -2.50) (-1.18, -1.37)  : AR

 ( 3.93, -6.14) ( 1.48,  6.58)
 ( 6.17,  9.42) ( 4.65, -4.75)
 (-7.17,-21.83) (-4.91,  2.29)
 ( 1.99,-14.38) ( 7.64,-10.79)  : B
```

10.3 Program Results

```
 F07WSF Example Program Results

 Solution(s)
                    1                   2
 1  ( 1.0000,-1.0000) (-1.0000, 2.0000)
 2  (-0.0000, 3.0000) ( 3.0000,-4.0000)
 3  (-4.0000,-5.0000) (-2.0000, 3.0000)
 4  ( 2.0000, 1.0000) ( 4.0000,-5.0000)
```