NAG Fortran Library Routine Document

F08YYF (ZTGSNA)

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

F08YYF (ZTGSNA) estimates condition numbers for specified eigenvalues and/or eigenvectors of a complex matrix pair in generalized Schur form.

2 Specification

```
      SUBROUTINE F08YYF (JOB, HOWMNY, SELECT, N, A, LDA, B, LDB, VL, LDVL, VR,
     1                   LDVR, S, DIF, MM, M, WORK, LWORK, IWORK, INFO)
      INTEGER            N, LDA, LDB, LDVL, LDVR, MM, M, LWORK, IWORK(*), INFO
      double precision   S(*), DIF(*)
      complex*16         A(LDA,*), B(LDB,*), VL(LDVL,*), VR(LDVR,*), WORK(*)
      LOGICAL            SELECT(*)
      CHARACTER*1        JOB, HOWMNY
```

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

3 Description

F08YYF (ZTGSNA) estimates condition numbers for specified eigenvalues and/or right eigenvectors of an n by n matrix pair (S,T) in generalized Schur form. The routine actually returns estimates of the reciprocals of the condition numbers in order to avoid possible overflow.

The pair (S,T) are in generalized Schur form if S and T are upper triangular as returned, for example, by F08XNF (ZGGES), or F08XSF (ZHGEQZ) with JOB = 'S'. The diagonal elements define the generalized eigenvalues, for $(\alpha_i,\beta_i), i=1,2,\ldots,n$, of the pair (S,T) and the eigenvalues are given by

$$\lambda_i=\alpha_i/\beta_i,$$

so that

$$\beta_i S x_i=\alpha_i T x_i \quad \text{or} \quad S x_i=\lambda_i T x_i,$$

where x_i is the corresponding (right) eigenvector.

If S and T are the result of a generalized Schur factorization of a matrix pair (A,B)

$$A=QSZ^{\mathrm{H}}, \quad B=QTZ^{\mathrm{H}}$$

then the eigenvalues and condition numbers of the pair (S,T) are the same as those of the pair (A,B).

Let $(\alpha,\beta)\neq(0,0)$ be a simple generalized eigenvalue of (A,B). Then the reciprocal of the condition number of the eigenvalue $\lambda=\alpha/\beta$ is defined as

$$s(\lambda)=\frac{\left(\left|y^{\mathrm{H}}Ax\right|^2+\left|y^{\mathrm{H}}Bx\right|^2\right)^{1/2}}{\left(\|x\|_2\|y\|_2\right)},$$

where x and y are the right and left eigenvectors of (A,B) corresponding to λ. If both α and β are zero, then (A,B) is singular and $s(\lambda)=-1$ is returned.

If U and V are unitary transformations such that

$$U^{\mathrm{H}}(A,B)V=(S,T)=\begin{pmatrix}\alpha & * \\ 0 & S_{22}\end{pmatrix}\begin{pmatrix}\beta & * \\ 0 & T_{22}\end{pmatrix},$$

where S_{22} and T_{22} are $(n-1)$ by $(n-1)$ matrices, then the reciprocal condition number is given by

$$\mathrm{Dif}(x) \equiv \mathrm{Dif}(y) = \mathrm{Dif}((\alpha, \beta), (S_{22}, T_{22})) = \sigma_{\min}(Z),$$

where $\sigma_{\min}(Z)$ denotes the smallest singular value of the $2(n-1)$ by $2(n-1)$ matrix

$$Z = \begin{pmatrix} \alpha \otimes I & -1 \otimes S_{22} \\ \beta \otimes I & -1 \otimes T_{22} \end{pmatrix}$$

and $\otimes$ is the Kronecker product.

See Anderson *et al.* (1999) (Sections 2.4.8 and 4.11) and Kågström and Poromaa (1996) for further details and information.

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

Kågström B and Poromaa P (1996) LAPACK-style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs *ACM Trans. Math. Software* **22** 78–103

5 Parameters

1: JOB – CHARACTER*1 *Input*

On entry: indicates whether condition numbers are required for eigenvalues and/or eigenvectors.

JOB = 'E'

Condition numbers for eigenvalues only are computed.

JOB = 'V'

Condition numbers for eigenvectors only are computed.

JOB = 'B'

Condition numbers for both eigenvalues and eigenvectors are computed.

Constraint: JOB = 'E', 'V' or 'B'.

2: HOWMNY – CHARACTER*1 *Input*

On entry: indicates how many condition numbers are to be computed.

HOWMNY = 'A'

Condition numbers for all eigenpairs are computed.

HOWMNY = 'S'

Condition numbers for selected eigenpairs (as specified by SELECT) are computed.

Constraint: HOWMNY = 'A' or 'S'.

3: SELECT($*$) – LOGICAL array *Input*

Note: the dimension of the array SELECT must be at least $\max(1, \mathrm{N})$ if HOWMNY = 'S' and at least 1 otherwise.

On entry: specifies the eigenpairs for which condition numbers are to be computed if HOWMNY = 'S'. To select condition numbers for the eigenpair corresponding to the eigenvalue λ_j, SELECT(j) must be set to .TRUE..

If HOWMNY = 'A', SELECT is not referenced.

4: N – INTEGER *Input*

On entry: n, the order of the matrix pair (S, T).

Constraint: $N \geq 0$.

5: A(LDA,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array A must be at least $\max(1, N)$.

On entry: the upper triangular matrix S.

6: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08YYF (ZTGSNA) is called.

Constraint: $LDA \geq \max(1, N)$.

7: B(LDB,$*$) – ***complex*16*** array *Input*

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

On entry: the upper triangular matrix T.

8: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08YYF (ZTGSNA) is called.

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

9: VL(LDVL,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array VL must be at least $\max(1, M)$.

On entry: if JOB = 'E' or 'B', VL must contain left eigenvectors of (S, T), corresponding to the eigenpairs specified by HOWMNY and SELECT. The eigenvectors must be stored in consecutive columns of VL, as returned by F08YXF (ZTGEVC).

If JOB = 'V', VL is not referenced.

10: LDVL – INTEGER *Input*

On entry: the first dimension of the array VL as declared in the (sub)program from which F08YYF (ZTGSNA) is called.

Constraints:

if JOB = 'E' or 'B', $LDVL \geq N$;
$LDVL \geq 1$ otherwise.

11: VR(LDVR,$*$) – ***complex*16*** array *Input*

Note: the second dimension of the array VR must be at least $\max(1, M)$.

On entry: if JOB = 'E' or 'B', VR must contain right eigenvectors of (S, T), corresponding to the eigenpairs specified by HOWMNY and SELECT. The eigenvectors must be stored in consecutive columns of VR, as returned by F08YXF (ZTGEVC) or F08WNF (ZGGEV).

If JOB = 'V', VR is not referenced.

12: LDVR – INTEGER *Input*

On entry: the first dimension of the array VR as declared in the (sub)program from which F08YYF (ZTGSNA) is called.

Constraints:

if JOB = 'E' or 'B', LDVR $\geq$ N;
LDVR ≥ 1 otherwise.

13: S($*$) – ***double precision*** array *Output*

Note: the dimension of the array S must be at least $\max(1, \text{MM})$.

On exit: if JOB = 'E' or 'B', the reciprocal condition numbers of the selected eigenvalues, stored in consecutive elements of the array.

If JOB = 'V', S is not referenced.

14: DIF($*$) – ***double precision*** array *Output*

Note: the dimension of the array DIF must be at least $\max(1, \text{MM})$.

On exit: if JOB = 'V' or 'B', the estimated reciprocal condition numbers of the selected eigenvectors, stored in consecutive elements of the array. If the eigenvalues cannot be reordered to compute DIF(j), DIF(j) is set to 0; this can only occur when the true value would be very small anyway.

If JOB = 'E', DIF is not referenced.

15: MM – INTEGER *Input*

On entry: the number of elements in the arrays S and DIF.

Constraint: MM $\geq$ M.

16: M – INTEGER *Output*

On exit: the number of elements of the arrays S and DIF used to store the specified condition numbers; for each selected eigenvalue one element is used. If HOWMNY = 'A', M is set to N.

17: WORK($*$) – ***complex*16*** array *Workspace*

Note: the dimension of the array WORK must be at least $\max(1, \text{LWORK})$.

On exit: if INFO = 0, WORK(1) returns the minimum LWORK.

18: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08YYF (ZTGSNA) is called.

If LWORK $= -1$, a workspace query is assumed; the routine only calculates the minimum size of the WORK array, returns this value as the first entry of the WORK array, and no error message related to LWORK is issued.

Constraints:

if LWORK $\neq -1$,
if JOB = 'V' or 'B', LWORK $\geq \max(1, 2 \times \text{N} \times \text{N})$;
LWORK $\geq \max(1, \text{N})$ otherwise.

19: IWORK($*$) – INTEGER array *Workspace*

Note: the dimension of the array IWORK must be at least $(\text{N} + 2)$.

If JOB = 'E', IWORK is not referenced.

20: 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 parameter had an illegal value. An explanatory message is output, and execution of the program is terminated.

7 Accuracy

None.

8 Further Comments

An approximate asymptotic error bound on the chordal distance between the computed eigenvalue $\tilde{\lambda}$ and the corresponding exact eigenvalue λ is

$$\chi\big(\tilde{\lambda}, \lambda\big) \leq \epsilon \|(A, B)\|_F / S(\lambda)$$

where ϵ is the ***machine precision***.

An approximate asymptotic error bound for the right or left computed eigenvectors $\tilde{x}$ or $\tilde{y}$ corresponding to the right and left eigenvectors x and y is given by

$$\theta(\tilde{z}, z) \leq \epsilon \|(A, B)\|_F / \mathrm{Dif}.$$

The real analogue of this routine is F08YLF (DTGSNA).

9 Example

To estimate condition numbers and approximate error estimates for all the eigenvalues and eigenvalues and right eigenvectors of the pair (S, T) given by

$$S = \begin{pmatrix} 4.0 + 4.0i & 1.0 + 1.0i & 1.0 + 1.0i & 2.0 - 1.0i \\ 0 & 2.0 + 1.0i & 1.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 2.0 - 1.0i & 1.0 + 1.0i \\ 0 & 0 & 0 & 6.0 - 2.0i \end{pmatrix}$$

and

$$T = \begin{pmatrix} 2.0 & 1.0 + 1.0i & 1.0 + 1.0i & 3.0 - 1.0i \\ 0 & 1.0 & 2.0 + 1.0i & 1.0 + 1.0i \\ 0 & 0 & 1.0 & 1.0 + 1.0i \\ 0 & 0 & 0 & 2.0 \end{pmatrix}.$$

The eigenvalues and eigenvectors are computed by calling F08YXF (ZTGEVC).

9.1 Program Text

```
*     F08YYF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=8)
      INTEGER          LDS, LDT, LDVL, LDVR, LWORK
      PARAMETER        (LDS=NMAX,LDT=NMAX,LDVL=NMAX,LDVR=NMAX,
     +                 LWORK=2*NMAX*NMAX)
*     .. Local Scalars ..
      DOUBLE PRECISION EPS, SNORM, STNRM, TNORM
      INTEGER          I, INFO, J, M, N
*     .. Local Arrays ..
      COMPLEX *16      S(LDS,NMAX), T(LDT,NMAX), VL(LDVL,NMAX),
```

```
     +                 VR(LDVR,NMAX), WORK(LWORK)
      DOUBLE PRECISION DIF(NMAX), RWORK(2*NMAX), SCON(NMAX)
      INTEGER          IWORK(NMAX+2)
      LOGICAL          SELECT(1)
*     .. External Functions ..
      DOUBLE PRECISION F06BNF, F06UAF, X02AJF
      EXTERNAL         F06BNF, F06UAF, X02AJF
*     .. External Subroutines ..
      EXTERNAL         ZTGEVC, ZTGSNA
*     .. Executable Statements ..
      WRITE (NOUT,*) 'F08YYF Example Program Results'
      WRITE (NOUT,*)
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N
      IF (N.LE.NMAX) THEN
*
*        Read S and T from data file
*
         READ (NIN,*) ((S(I,J),J=1,N),I=1,N)
         READ (NIN,*) ((T(I,J),J=1,N),I=1,N)
*
*        Calculate the left and right generalized eigenvectors of the
*        pair (S,T).
*
         CALL ZTGEVC('Both','All',SELECT,N,S,LDS,T,LDT,VL,LDVL,VR,LDVR,
     +               N,M,WORK,RWORK,INFO)
*
*        Estimate condition numbers for all the generalized eigenvalues
*        and right eigenvectors of the pair (S,T)
*
         CALL ZTGSNA('Both','All',SELECT,N,S,LDS,T,LDT,VL,LDVL,VR,LDVR,
     +               SCON,DIF,N,M,WORK,LWORK,IWORK,INFO)
*
*        Print condition numbers of eigenvalues and right eigenvectors
*
         WRITE (NOUT,*) 'SCON'
         WRITE (NOUT,99999) (SCON(I),I=1,M)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'DIF'
         WRITE (NOUT,99999) (DIF(I),I=1,M)
*
*        Calculate approximate error estimates
*
*        Compute the 1-norms of S and T and then compute
*        SQRT(SNORM**2 + TNORM**2)
*
         EPS = X02AJF()
         SNORM = F06UAF('1-norm',N,N,S,LDS,RWORK)
         TNORM = F06UAF('1-norm',N,N,T,LDT,RWORK)
         STNRM = F06BNF(SNORM,TNORM)
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Approximate error estimates for eigenvalues of (S,T)'
         WRITE (NOUT,99999) (EPS*TNORM/SCON(I),I=1,M)
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Approximate error estimates for right eigenvectors of (S,T)'
         WRITE (NOUT,99999) (EPS*TNORM/DIF(I),I=1,M)
      ELSE
         WRITE (NOUT,*) 'NMAX too small'
      END IF
      STOP
*
99999 FORMAT ((3X,1P,7E11.1))
      END
```

9.2 Program Data

```
F08YYF Example Program Data
  4                                                     :Value of N
 ( 4.0, 4.0) ( 1.0, 1.0) ( 1.0, 1.0) ( 2.0,-1.0)
 ( 0.0, 0.0) ( 2.0, 1.0) ( 1.0, 1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 2.0,-1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 6.0,-2.0)   :End of matrix S
 ( 2.0, 0.0) ( 1.0, 1.0) ( 1.0, 1.0) ( 3.0,-1.0)
 ( 0.0, 0.0) ( 1.0, 0.0) ( 2.0, 1.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 1.0, 0.0) ( 1.0, 1.0)
 ( 0.0, 0.0) ( 0.0, 0.0) ( 0.0, 0.0) ( 2.0, 0.0)   :End of matrix T
```

9.3 Program Results

```
 F08YYF Example Program Results

 SCON
       1.0E+00    8.2E-01    7.2E-01    8.2E-01

 DIF
       3.2E-01    3.6E-01    5.5E-01    2.8E-01

 Approximate error estimates for eigenvalues of (S,T)
       8.5E-16    1.1E-15    1.2E-15    1.1E-15

 Approximate error estimates for right eigenvectors of (S,T)
       2.7E-15    2.5E-15    1.6E-15    3.2E-15
```
