NAG Library Routine Document

F08SEF (DSYGST)

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

F08SEF (DSYGST) reduces a real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, where A is a real symmetric matrix and B has been factorized by F07FDF (DPOTRF).

2 Specification

```
SUBROUTINE F08SEF (ITYPE, UPLO, N, A, LDA, B, LDB, INFO)

INTEGER            ITYPE, N, LDA, LDB, INFO
REAL (KIND=nag_wp) A(LDA,*), B(LDB,*)
CHARACTER(1)       UPLO
```

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

3 Description

To reduce the real symmetric-definite generalized eigenproblem $Az = \lambda Bz$, $ABz = \lambda z$ or $BAz = \lambda z$ to the standard form $Cy = \lambda y$, F08SEF (DSYGST) must be preceded by a call to F07FDF (DPOTRF) which computes the Cholesky factorization of B; B must be positive definite.

The different problem types are specified by the parameter ITYPE, as indicated in the table below. The table shows how C is computed by the routine, and also how the eigenvectors z of the original problem can be recovered from the eigenvectors of the standard form.

ITYPE	Problem	UPLO	B	C	z
1	$Az = \lambda Bz$	'U' 'L'	$U^{\mathrm{T}}U$ LL^{T}	$U^{-\mathrm{T}}AU^{-1}$ $L^{-1}AL^{-\mathrm{T}}$	$U^{-1}y$ $L^{-\mathrm{T}}y$
2	$ABz = \lambda z$	'U' 'L'	$U^{\mathrm{T}}U$ LL^{T}	UAU^{T} $L^{\mathrm{T}}AL$	$U^{-1}y$ $L^{-\mathrm{T}}y$
3	$BAz = \lambda z$	'U' 'L'	$U^{\mathrm{T}}U$ LL^{T}	UAU^{T} $L^{\mathrm{T}}AL$	$U^{\mathrm{T}}y$ Ly

4 References

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

5 Parameters

1: ITYPE – INTEGER *Input*

On entry: indicates how the standard form is computed.

ITYPE = 1

if UPLO = 'U', $C = U^{-\mathrm{T}} A U^{-1}$;

if UPLO = 'L', $C = L^{-1} A L^{-\mathrm{T}}$.

ITYPE = 2 or 3

if UPLO = 'U', $C = U A U^{\mathrm{T}}$;

if UPLO = 'L', $C = L^{\mathrm{T}} A L$.

Constraint: ITYPE = 1, 2 or 3.

2: UPLO – CHARACTER(1) *Input*

On entry: indicates whether the upper or lower triangular part of A is stored and how B has been factorized.

UPLO = 'U'
: The upper triangular part of A is stored and $B = U^{\mathrm{T}} U$.

UPLO = 'L'
: The lower triangular part of A is stored and $B = L L^{\mathrm{T}}$.

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

3: N – INTEGER *Input*

On entry: n, the order of the matrices A and B.

Constraint: N ≥ 0.

4: A(LDA,$*$) – REAL (KIND=nag_wp) array *Input/Output*

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

On entry: the n by n symmetric matrix A.

If UPLO = 'U', the upper triangular part of A must be stored and the elements of the array below the diagonal are not referenced.

If UPLO = 'L', the lower triangular part of A must be stored and the elements of the array above the diagonal are not referenced.

On exit: the upper or lower triangle of A is overwritten by the corresponding upper or lower triangle of C as specified by ITYPE and UPLO.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08SEF (DSYGST) is called.

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

6: B(LDB,$*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: the Cholesky factor of B as specified by UPLO and returned by F07FDF (DPOTRF).

7: LDB – INTEGER *Input*

On entry: the first dimension of the array B as declared in the (sub)program from which F08SEF (DSYGST) is called.

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

8: INFO – INTEGER *Output*

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

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

$\text{INFO} < 0$

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

7 Accuracy

Forming the reduced matrix C is a stable procedure. However it involves implicit multiplication by B^{-1} (if $\text{ITYPE} = 1$) or B (if $\text{ITYPE} = 2$ or 3). When F08SEF (DSYGST) is used as a step in the computation of eigenvalues and eigenvectors of the original problem, there may be a significant loss of accuracy if B is ill-conditioned with respect to inversion. See the document for F08SAF (DSYGV) for further details.

8 Further Comments

The total number of floating point operations is approximately n^3.

The complex analogue of this routine is F08SSF (ZHEGST).

9 Example

This example computes all the eigenvalues of $Az = \lambda Bz$, where

$$A = \begin{pmatrix} 0.24 & 0.39 & 0.42 & -0.16 \\ 0.39 & -0.11 & 0.79 & 0.63 \\ 0.42 & 0.79 & -0.25 & 0.48 \\ -0.16 & 0.63 & 0.48 & -0.03 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} 4.16 & -3.12 & 0.56 & -0.10 \\ -3.12 & 5.03 & -0.83 & 1.09 \\ 0.56 & -0.83 & 0.76 & 0.34 \\ -0.10 & 1.09 & 0.34 & 1.18 \end{pmatrix}.$$

Here B is symmetric positive definite and must first be factorized by F07FDF (DPOTRF). The program calls F08SEF (DSYGST) to reduce the problem to the standard form $Cy = \lambda y$; then F08FEF (DSYTRD) to reduce C to tridiagonal form, and F08JFF (DSTERF) to compute the eigenvalues.

9.1 Program Text

```
    Program f08sefe

!     F08SEF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: dpotrf, dsterf, dsygst, dsytrd, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, info, lda, ldb, lwork, n
      Character (1)                    :: uplo
```

```
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), b(:,:), d(:), e(:), tau(:),  &
                                          work(:)
!     .. Executable Statements ..
      Write (nout,*) 'F08SEF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      ldb = n
      lwork = 64*n
      Allocate (a(lda,n),b(ldb,n),d(n),e(n-1),tau(n),work(lwork))

!     Read A and B from data file

      Read (nin,*) uplo
      If (uplo=='U') Then
        Read (nin,*)(a(i,i:n),i=1,n)
        Read (nin,*)(b(i,i:n),i=1,n)
      Else If (uplo=='L') Then
        Read (nin,*)(a(i,1:i),i=1,n)
        Read (nin,*)(b(i,1:i),i=1,n)
      End If

!     Compute the Cholesky factorization of B
!     The NAG name equivalent of dpotrf is f07fdf
      Call dpotrf(uplo,n,b,ldb,info)

      Write (nout,*)
      If (info>0) Then
        Write (nout,*) 'B is not positive definite.'
      Else

!       Reduce the problem to standard form C*y = lambda*y, storing
!       the result in A
!       The NAG name equivalent of dsygst is f08sef
        Call dsygst(1,uplo,n,a,lda,b,ldb,info)

!       Reduce C to tridiagonal form T = (Q**T)*C*Q
!       The NAG name equivalent of dsytrd is f08fef
        Call dsytrd(uplo,n,a,lda,d,e,tau,work,lwork,info)

!       Calculate the eigenvalues of T (same as C)
!       The NAG name equivalent of dsterf is f08jff
        Call dsterf(n,d,e,info)

        If (info>0) Then
          Write (nout,*) 'Failure to converge.'
        Else

!         Print eigenvalues

          Write (nout,*) 'Eigenvalues'
          Write (nout,99999) d(1:n)
        End If
      End If

99999 Format (3X,(9F8.4))
    End Program f08sefe
```

9.2 Program Data

```
F08SEF Example Program Data
  4                            :Value of N
  'L'                          :Value of UPLO
  0.24
  0.39  -0.11
  0.42   0.79  -0.25
```

```
-0.16   0.63   0.48  -0.03   :End of matrix A
 4.16
-3.12   5.03
 0.56  -0.83   0.76
-0.10   1.09   0.34   1.18   :End of matrix B
```

9.3 Program Results

```
F08SEF Example Program Results

Eigenvalues
   -2.2254 -0.4548  0.1001  1.1270
```