

NAG Library Routine Document

F08SAF (DSYGV)

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

F08SAF (DSYGV) computes all the eigenvalues and, optionally, the eigenvectors of a real generalized symmetric-definite eigenproblem, of the form

$$Az = \lambda Bz, \quad ABz = \lambda z \quad \text{or} \quad BAz = \lambda z,$$

where A and B are symmetric and B is also positive definite.

2 Specification

```
SUBROUTINE F08SAF (ITYPE, JOBZ, UPLO, N, A, LDA, B, LDB, W, WORK, LWORK,      &
                  INFO)
INTEGER                ITYPE, N, LDA, LDB, LWORK, INFO
REAL (KIND=nag_wp)    A(LDA,*), B(LDB,*), W(N), WORK(max(1,LWORK))
CHARACTER(1)          JOBZ, UPLO
```

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

3 Description

F08SAF (DSYGV) first performs a Cholesky factorization of the matrix B as $B = U^T U$, when $UPLO = 'U'$ or $B = LL^T$, when $UPLO = 'L'$. The generalized problem is then reduced to a standard symmetric eigenvalue problem

$$Cx = \lambda x,$$

which is solved for the eigenvalues and, optionally, the eigenvectors; the eigenvectors are then backtransformed to give the eigenvectors of the original problem.

For the problem $Az = \lambda Bz$, the eigenvectors are normalized so that the matrix of eigenvectors, z , satisfies

$$Z^T A Z = \Lambda \quad \text{and} \quad Z^T B Z = I,$$

where Λ is the diagonal matrix whose diagonal elements are the eigenvalues. For the problem $ABz = \lambda z$ we correspondingly have

$$Z^{-1} A Z^{-T} = \Lambda \quad \text{and} \quad Z^T B Z = I,$$

and for $BAz = \lambda z$ we have

$$Z^T A Z = \Lambda \quad \text{and} \quad Z^T B^{-1} Z = I.$$

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 Arguments

- 1: ITYPE – INTEGER *Input*
On entry: specifies the problem type to be solved.
 ITYPE = 1
 $Az = \lambda Bz$.
 ITYPE = 2
 $ABz = \lambda z$.
 ITYPE = 3
 $BAz = \lambda z$.
Constraint: ITYPE = 1, 2 or 3.
- 2: JOBZ – CHARACTER(1) *Input*
On entry: indicates whether eigenvectors are computed.
 JOBZ = 'N'
 Only eigenvalues are computed.
 JOBZ = 'V'
 Eigenvalues and eigenvectors are computed.
Constraint: JOBZ = 'N' or 'V'.
- 3: UPLO – CHARACTER(1) *Input*
On entry: if UPLO = 'U', the upper triangles of A and B are stored.
 If UPLO = 'L', the lower triangles of A and B are stored.
Constraint: UPLO = 'U' or 'L'.
- 4: N – INTEGER *Input*
On entry: n , the order of the matrices A and B .
Constraint: $N \geq 0$.
- 5: A(LDA,*) – REAL (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array A must be at least $\max(1, N)$.
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: if JOBZ = 'V', A contains the matrix Z of eigenvectors. The eigenvectors are normalized as follows:
 if ITYPE = 1 or 2, $Z^T B Z = I$;
 if ITYPE = 3, $Z^T B^{-1} Z = I$.
 If JOBZ = 'N', the upper triangle (if UPLO = 'U') or the lower triangle (if UPLO = 'L') of A , including the diagonal, is overwritten.

- 6: LDA – INTEGER *Input*
On entry: the first dimension of the array A as declared in the (sub)program from which F08SAF (DSYGV) is called.
Constraint: $LDA \geq \max(1, N)$.
- 7: B(LDB, *) – REAL (KIND=nag_wp) array *Input/Output*
Note: the second dimension of the array B must be at least $\max(1, N)$.
On entry: the n by n symmetric positive definite matrix B .
 If UPLO = 'U', the upper triangular part of B must be stored and the elements of the array below the diagonal are not referenced.
 If UPLO = 'L', the lower triangular part of B must be stored and the elements of the array above the diagonal are not referenced.
On exit: if $0 \leq \text{INFO} \leq N$, the part of B containing the matrix is overwritten by the triangular factor U or L from the Cholesky factorization $B = U^T U$ or $B = L L^T$.
- 8: LDB – INTEGER *Input*
On entry: the first dimension of the array B as declared in the (sub)program from which F08SAF (DSYGV) is called.
Constraint: $LDB \geq \max(1, N)$.
- 9: W(N) – REAL (KIND=nag_wp) array *Output*
On exit: the eigenvalues in ascending order.
- 10: WORK(max(1, LWORK)) – REAL (KIND=nag_wp) array *Workspace*
On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimal performance.
- 11: LWORK – INTEGER *Input*
On entry: the dimension of the array WORK as declared in the (sub)program from which F08SAF (DSYGV) is called.
 If LWORK = -1, a workspace query is assumed; the routine only calculates the optimal 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.
 Suggested value: for optimal performance, $LWORK \geq (nb + 2) \times N$, where nb is the optimal **block size** for F08FEF (DSYTRD).
Constraint: $LWORK \geq \max(1, 3 \times N - 1)$.
- 12: 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.

INFO = 1 to N

If INFO = i , F08FAF (DSYEV) failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

INFO > N

F07FDF (DPOTRF) returned an error code; i.e., if INFO = $N + i$, for $1 \leq i \leq N$, then the leading minor of order i of B is not positive definite. The factorization of B could not be completed and no eigenvalues or eigenvectors were computed.

7 Accuracy

If B is ill-conditioned with respect to inversion, then the error bounds for the computed eigenvalues and vectors may be large, although when the diagonal elements of B differ widely in magnitude the eigenvalues and eigenvectors may be less sensitive than the condition of B would suggest. See Section 4.10 of Anderson *et al.* (1999) for details of the error bounds.

The example program below illustrates the computation of approximate error bounds.

8 Parallelism and Performance

F08SAF (DSYGV) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

F08SAF (DSYGV) 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 floating-point operations is proportional to n^3 .

The complex analogue of this routine is F08SNF (ZHEGV).

10 Example

This example finds all the eigenvalues and eigenvectors of the generalized symmetric eigenproblem $Az = \lambda Bz$, where

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

together with an estimate of the condition number of B , and approximate error bounds for the computed eigenvalues and eigenvectors.

The example program for F08SCF (DSYGVD) illustrates solving a generalized symmetric eigenproblem of the form $ABz = \lambda z$.

10.1 Program Text

```

Program f08safe

!      F08SAF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
!      Use nag_library, Only: blas_damax_val, ddisna, dsygv, dtrcon, f06rcf,      &
!                               nag_wp, x02ajf, x04caf
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Real (Kind=nag_wp), Parameter      :: zero = 0.0_nag_wp
!      Integer, Parameter                  :: nb = 64, nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)                  :: anorm, bnorm, eps, r, rcond, rcondb, &
!                                          t1, t2, t3
!      Integer                             :: i, ifail, info, k, lda, ldb, lwork, &
!                                          n
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable     :: a(:,,:), b(:,,:), eerbnd(:),      &
!                                          rcondz(:), w(:), work(:), zerbnd(:)
!      Real (Kind=nag_wp)                  :: dummy(1)
!      Integer, Allocatable                 :: iwork(:)
!      .. Intrinsic Procedures ..
!      Intrinsic                           :: abs, max, nint
!      .. Executable Statements ..
!      Write (nout,*) 'F08SAF Example Program Results'
!      Write (nout,*)
!      Skip heading in data file
!      Read (nin,*)
!      Read (nin,*) n
!      lda = n
!      ldb = n
!      Allocate (a(lda,n),b(ldb,n),eerbnd(n),rcondz(n),w(n),zerbnd(n),iwork(n))

!      Use routine workspace query to get optimal workspace.
!      lwork = -1
!      The NAG name equivalent of dsygv is f08saf
!      Call dsygv(1,'Vectors','Upper',n,a,lda,b,ldb,w,dummy,lwork,info)

!      Make sure that there is enough workspace for block size nb.
!      lwork = max((nb+2)*n,nint(dummy(1)))
!      Allocate (work(lwork))

!      Read the upper triangular parts of the matrices A and B

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

!      Compute the one-norms of the symmetric matrices A and B

!      anorm = f06rcf('One norm','Upper',n,a,lda,work)
!      bnorm = f06rcf('One norm','Upper',n,b,ldb,work)

!      Solve the generalized symmetric eigenvalue problem
!      A*x = lambda*B*x (ITYPE = 1)

!      The NAG name equivalent of dsygv is f08saf
!      Call dsygv(1,'Vectors','Upper',n,a,lda,b,ldb,w,work,lwork,info)

!      If (info==0) Then

!      Print solution

!      Write (nout,*) 'Eigenvalues'
!      Write (nout,99999) w(1:n)

!      Write (nout,*)

```

```

      Flush (nout)

!      Normalize the eigenvectors, largest positive
      Do i = 1, n
        Call blas_damax_val(n,a(1,i),1,k,r)
        If (a(k,i)<zero) Then
          a(1:n,i) = -a(1:n,i)
        End If
      End Do

!      ifail: behaviour on error exit
!      =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
      ifail = 0
      Call x04caf('General',' ',n,n,a,lda,'Eigenvectors',ifail)

!      Call DTRCON (F07TGF) to estimate the reciprocal condition
!      number of the Cholesky factor of B. Note that:
!      cond(B) = 1/rcond**2

!      The NAG name equivalent of dtrcon is f07tgf
      Call dtrcon('One norm','Upper','Non-unit',n,b,ldb,rcond,work,iwork,      &
        info)

!      Print the reciprocal condition number of B

      rcondb = rcond**2
      Write (nout,*)
      Write (nout,*) 'Estimate of reciprocal condition number for B'
      Write (nout,99998) rcondb
      Flush (nout)

!      Get the machine precision, eps, and if rcondb is not less
!      than eps**2, compute error estimates for the eigenvalues and
!      eigenvectors

      eps = x02ajf()
      If (rcond>=eps) Then

!      Call DDISNA (F08FLF) to estimate reciprocal condition
!      numbers for the eigenvectors of (A - lambda*B)

      Call ddisna('Eigenvectors',n,n,w,rcondz,info)

!      Compute the error estimates for the eigenvalues and
!      eigenvectors

      t1 = eps/rcondb
      t2 = anorm/bnorm
      t3 = t2/rcond
      Do i = 1, n
        eerbnd(i) = t1*(t2+abs(w(i)))
        zerbnd(i) = t1*(t3+abs(w(i)))/rcondz(i)
      End Do

!      Print the approximate error bounds for the eigenvalues
!      and vectors

      Write (nout,*)
      Write (nout,*) 'Error estimates for the eigenvalues'
      Write (nout,99998) eerbnd(1:n)
      Write (nout,*)
      Write (nout,*) 'Error estimates for the eigenvectors'
      Write (nout,99998) zerbnd(1:n)
    Else
      Write (nout,*)
      Write (nout,*) 'B is very ill-conditioned, error ',      &
        'estimates have not been computed'
    End If
  Else If (info>n .And. info<=2*n) Then
    i = info - n
    Write (nout,99997) 'The leading minor of order ', i,      &

```

```

      ' of B is not positive definite'
Else
  Write (nout,99996) 'Failure in DSYGV. INFO =', info
End If

99999 Format (3X,(6F11.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4,A)
99996 Format (1X,A,I4)
End Program f08safe

```

10.2 Program Data

F08SAF Example Program Data

```

4                               :Value of N

0.24   0.39   0.42  -0.16
      -0.11   0.79   0.63
          -0.25   0.48
          -0.03 :End of matrix A

4.16  -3.12   0.56  -0.10
      5.03  -0.83   1.09
          0.76   0.34
          1.18 :End of matrix B

```

10.3 Program Results

F08SAF Example Program Results

```

Eigenvalues
-2.2254   -0.4548   0.1001   1.1270

Eigenvectors
      1      2      3      4
1      0.0690  -0.3080  -0.4469   0.5528
2      0.5740  -0.5329  -0.0371   0.6766
3      1.5428   0.3496   0.0505   0.9276
4     -1.4004   0.6211   0.4743  -0.2510

Estimate of reciprocal condition number for B
5.8E-03

Error estimates for the eigenvalues
4.7E-14   1.2E-14   5.6E-15   2.5E-14

Error estimates for the eigenvectors
5.2E-14   1.0E-13   9.2E-14   6.9E-14

```
