NAG Library Routine Document

F08FAF (DSYEV)

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

F08FAF (DSYEV) computes all the eigenvalues and, optionally, all the eigenvectors of a real n by n symmetric matrix A.

2 Specification

```
SUBROUTINE F08FAF (JOBZ, UPLO, N, A, LDA, W, WORK, LWORK, INFO)

INTEGER             N, LDA, LWORK, INFO
REAL (KIND=nag_wp) A(LDA,*), W(N), WORK(max(1,LWORK))
CHARACTER(1)        JOBZ, UPLO
```

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

3 Description

The symmetric matrix A is first reduced to tridiagonal form, using orthogonal similarity transformations, and then the QR algorithm is applied to the tridiagonal matrix to compute the eigenvalues and (optionally) the eigenvectors.

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: 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'.

2: UPLO – CHARACTER(1) *Input*

On entry: if UPLO = 'U', the upper triangular part of A is stored.

If UPLO = 'L', the lower triangular part of A is stored.

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

3: N – INTEGER *Input*

On entry: n, the order of the matrix A.

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

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: if JOBZ = 'V', then A contains the orthonormal eigenvectors of the matrix A.

If JOBZ = 'N', then on exit the lower triangle (if UPLO = 'L') or the upper triangle (if UPLO = 'U') of A, including the diagonal, is overwritten.

5: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F08FAF (DSYEV) is called.

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

6: W(N) – REAL (KIND=nag_wp) array *Output*

On exit: the eigenvalues in ascending order.

7: WORK($\max(1, \mathrm{LWORK})$) – REAL (KIND=nag_wp) array *Workspace*

On exit: if INFO = 0, WORK(1) contains the minimum value of LWORK required for optimal performance.

8: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which F08FAF (DSYEV) is called.

If $\mathrm{LWORK} = -1$, 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, $\mathrm{LWORK} \geq (nb + 2) \times \mathrm{N}$, where nb is the optimal ***block size*** for F08FEF (DSYTRD).

Constraint: $\mathrm{LWORK} \geq \max(1, 3 \times \mathrm{N} - 1)$.

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

INFO > 0

If INFO $= i$, the algorithm failed to converge; i off-diagonal elements of an intermediate tridiagonal form did not converge to zero.

7 Accuracy

The computed eigenvalues and eigenvectors are exact for a nearby matrix $(A + E)$, where

$$\|E\|_2 = O(\epsilon)\|A\|_2,$$

and ϵ is the ***machine precision***. See Section 4.7 of Anderson *et al.* (1999) for further details.

8 Further Comments

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

The complex analogue of this routine is F08FNF (ZHEEV).

9 Example

This example finds all the eigenvalues and eigenvectors of the symmetric matrix

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 2 & 3 & 4 \\ 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 4 \end{pmatrix},$$

together with approximate error bounds for the computed eigenvalues and eigenvectors.

9.1 Program Text

```
    Program f08fafe

!     F08FAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: ddisna, dsyev, nag_wp, x02ajf, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nb = 64, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: eerrbd, eps
      Integer                          :: i, ifail, info, lda, lwork, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:,:), rcondz(:), w(:), work(:),    &
                                          zerrbd(:)
      Real (Kind=nag_wp)               :: dummy(1)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: abs, max, nint
!     .. Executable Statements ..
      Write (nout,*) 'F08FAF Example Program Results'
      Write (nout,*)
!     Skip heading in data file
      Read (nin,*)
      Read (nin,*) n
      lda = n
      Allocate (a(lda,n),rcondz(n),w(n),zerrbd(n))

!     Use routine workspace query to get optimal workspace.
!     The NAG name equivalent of dsyev is f08faf
      lwork = -1
      Call dsyev('Vectors','Upper',n,a,lda,w,dummy,lwork,info)
```

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

!     Read the upper triangular part of the matrix A from data file

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

!     Solve the symmetric eigenvalue problem
!     The NAG name equivalent of dsyev is f08faf
      Call dsyev('Vectors','Upper',n,a,lda,w,work,lwork,info)

      If (info==0) Then

!       Print solution

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

!       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)

!       Get the machine precision, EPS and compute the approximate
!       error bound for the computed eigenvalues.  Note that for
!       the 2-norm, max( abs(W(i)) ) = norm(A), and since the
!       eigenvalues are returned in ascending order
!       max( abs(W(i)) ) = max( abs(W(1)), abs(W(n)))

        eps = x02ajf()
        eerrbd = eps*max(abs(w(1)),abs(w(n)))

!       Call DDISNA (F08FLF) to estimate reciprocal condition
!       numbers for the eigenvectors
        Call ddisna('Eigenvectors',n,n,w,rcondz,info)

!       Compute the error estimates for the eigenvectors

        Do i = 1, n
          zerrbd(i) = eerrbd/rcondz(i)
        End Do

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

        Write (nout,*)
        Write (nout,*) 'Error estimate for the eigenvalues'
        Write (nout,99998) eerrbd
        Write (nout,*)
        Write (nout,*) 'Error estimates for the eigenvectors'
        Write (nout,99998) zerrbd(1:n)
      Else
        Write (nout,99997) 'Failure in DSYEV. INFO =', info
      End If

99999 Format (3X,(8F8.4))
99998 Format (4X,1P,6E11.1)
99997 Format (1X,A,I4)
    End Program f08fafe
```

9.2 Program Data

```
F08FAF Example Program Data

  4                 :Value of N

  1.0  2.0  3.0  4.0
       2.0  3.0  4.0
            3.0  4.0
                 4.0 :End of matrix A
```

9.3 Program Results

```
 F08FAF Example Program Results

 Eigenvalues
    -2.0531 -0.5146 -0.2943 12.8621
 Eigenvectors
          1       2       3       4
 1   0.7003 -0.5144 -0.2767  0.4103
 2   0.3592  0.4851  0.6634  0.4422
 3  -0.1569  0.5420 -0.6504  0.5085
 4  -0.5965 -0.4543  0.2457  0.6144

 Error estimate for the eigenvalues
        1.4E-15

 Error estimates for the eigenvectors
        9.3E-16    6.5E-15    6.5E-15    1.1E-16
```