NAG Toolbox for MATLAB

f02fa

1 Purpose

f02fa computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix.

2 Syntax

```
[a, w, ifail] = f02fa(job, uplo, a, 'n', n)
```

3 Description

f02fa computes all the eigenvalues, and optionally all the eigenvectors, of a real symmetric matrix A:

$$Az_i = \lambda_i z_i, \qquad i = 1, 2, \ldots, n.$$

In other words, it computes the spectral factorization of A:

$$A = Z \Lambda Z^{\mathrm{T}},$$

where Λ is a diagonal matrix whose diagonal elements are the eigenvalues λ_i, and Z is an orthogonal matrix, whose columns are the eigenvectors z_i.

4 References

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

Parlett B N 1998 *The Symmetric Eigenvalue Problem* SIAM, Philadelphia

5 Parameters

5.1 Compulsory Input Parameters

1: **job – string**

Indicates whether eigenvectors are to be computed.

job = 'N'

Only eigenvalues are computed.

job = 'V'

Eigenvalues and eigenvectors are computed.

Constraint: **job** = 'N' or 'V'.

2: **uplo – string**

Indicates whether the upper or lower triangular part of A is stored.

uplo = 'U'

The upper triangular part of A is stored.

uplo = 'L'

The lower triangular part of A is stored.

Constraint: **uplo** = 'U' or 'L'.

3: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

The n by n symmetric matrix A.

If **uplo** = 'U', the upper triangle of A must be stored and the elements of the array below the diagonal need not be set.

If **uplo** = 'L', the lower triangle of A must be stored and the elements of the array above the diagonal need not be set.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The dimension of the array **n**.

n, the order of the matrix A.

Constraint: $\mathbf{n} \geq 0$.

5.3 Input Parameters Omitted from the MATLAB Interface

lda, work, lwork

5.4 Output Parameters

1: **a(lda,$*$) – double array**

The first dimension of the array **a** must be at least $\max(1, \mathbf{n})$

The second dimension of the array must be at least $\max(1, \mathbf{n})$

If **job** = 'V', **a** contains the orthogonal matrix Z of eigenvectors, with the ith column holding the eigenvector z_i associated with the eigenvalue λ_i (stored in $\mathbf{w}(i)$).

If **uplo** = 'U', the upper triangular part of **a** is overwritten.

If **uplo** = 'L', the lower triangular part of **a** if overwritten.

2: **w($*$) – double array**

Note: the dimension of the array **w** must be at least $\max(1, \mathbf{n})$.

The eigenvalues in ascending order.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **job** $\neq$ 'N' or 'V',
or **uplo** $\neq$ 'U' or 'L',
or $\mathbf{n} < 0$,
or $\mathbf{lda} < \max(1, \mathbf{n})$,
or $\mathbf{lwork} < \max(1, 3 \times \mathbf{n})$.

ifail $= 2$

The *QR* algorithm failed to compute all the eigenvalues.

7 Accuracy

If λ_i is an exact eigenvalue, and $\tilde{\lambda}_i$ is the corresponding computed value, then

$$\left|\tilde{\lambda}_i - \lambda_i\right| \leq c(n)\epsilon\|A\|_2,$$

where $c(n)$ is a modestly increasing function of n, and ϵ is the ***machine precision***.

If z_i is the corresponding exact eigenvector, and $\tilde{z}_i$ is the corresponding computed eigenvector, then the angle $\theta(\tilde{z}_i, z_i)$ between them is bounded as follows:

$$\theta(\tilde{z}_i, z_i) \leq \frac{c(n)\epsilon\|A\|_2}{\min_{i \neq j}\left|\lambda_i - \lambda_j\right|}.$$

Thus the accuracy of a computed eigenvector depends on the gap between its eigenvalue and all the other eigenvalues.

8 Further Comments

f02fa calls functions from LAPACK in Chapter F08. It first reduces A to tridiagonal form T, using an orthogonal similarity transformation: $A = QTQ^{\mathrm{T}}$. If only eigenvalues are required, the function uses a root-free variant of the symmetric tridiagonal *QR* algorithm. If eigenvectors are required, the function first forms the orthogonal matrix Q that was used in the reduction to tridiagonal form; it then uses the symmetric tridiagonal *QR* algorithm to reduce T to Λ, using a further orthogonal transformation: $T = S\Lambda S^{\mathrm{T}}$; and at the same time accumulates the matrix $Z = QS$.

Each eigenvector z is normalized so that $\|z\|_2 = 1$ and the element of largest absolute value is positive.

The time taken by the function is approximately proportional to n^3.

9 Example

```
job = 'Vectors';
uplo = 'L';
a = [4.16, 0, 0, 0;
     -3.12, 5.03, 0, 0;
     0.56, -0.83, 0.76, 0;
     -0.1, 1.18, 0.34, 1.18];
[aOut, w, ifail] = f02fa(job, uplo, a)
```

```
aOut =
    0.1859   -0.4209    0.6230   -0.6325
    0.3791   -0.3108    0.4405    0.7521
    0.6621    0.7210    0.1588   -0.1288
   -0.6192    0.4543    0.6266    0.1329
w =
    0.1239
    1.0051
    1.9963
    8.0047
ifail =
          0
```