NAG Toolbox for MATLAB

f02ha

1 Purpose

f02ha computes all the eigenvalues, and optionally all the eigenvectors, of a complex Hermitian matrix.

2 Syntax

```
[a, w, ifail] = f02ha(job, uplo, a, 'n', n)
```

3 Description

f02ha computes all the eigenvalues, and optionally all the eigenvectors, of a complex Hermitian matrix A:

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

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

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

where Λ is a diagonal matrix whose diagonal elements are the eigenvalues λ_i, and Z is a unitary 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,*) – complex 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 Hermitian 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, rwork, work, lwork

5.4 Output Parameters

1: **a(lda,*) – complex 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 unitary 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 **job** = 'N', the original contents of **a** are 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** ≠ 'N' or 'V',
or **uplo** ≠ 'U' or 'L',
or $\mathbf{n} < 0$,
or $\mathbf{lda} < \max(1, \mathbf{n})$,
or $\mathbf{lwork} < \max(1, 2 \times \mathbf{n})$.

ifail = 2

The QR algorithm failed to compute all the eigenvalues.

ifail = 3

For some i, $\mathbf{a}(i,i)$ has a nonzero imaginary part (thus A is not Hermitian).

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

f02ha calls functions from LAPACK in Chapter F08. It first reduces A to real tridiagonal form T, using a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. 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 unitary 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 real orthogonal transformation: $T = S\Lambda S^{\mathrm{T}}$; and at the same time it accumulates the matrix $Z = QS$.

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

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

9 Example

```
job = 'Vectors';
uplo = 'L';
a = [complex(-2.28, 0), complex(0, 0), complex(0, 0), complex(0, 0);
       complex(1.78, 2.03), complex(-1.12, 0), complex(0, 0), complex(0,
0);
         complex(2.26, -0.1), complex(0.01, -0.43), complex(-0.37, 0),
complex(0, 0);
      complex(-0.12, -2.53), complex(-1.07, -0.86), complex(2.31, +0.92),
complex(-0.73, +0)];
[aOut, w, ifail] = f02ha(job, uplo, a)
```

```
aOut =
    0.7299              -0.2120 + 0.1497i    0.1000 - 0.3570i    0.1991 +
0.4720i
  -0.1663 - 0.2061i    0.7307                0.2863 - 0.3353i  -0.2467 +
0.3751i
  -0.4165 - 0.1417i  -0.3291 + 0.0479i    0.6890              0.4468 +
0.1466i
   0.1743 + 0.4162i   0.5200 + 0.1329i   0.0662 + 0.4347i   0.5612
w =
   -6.0002
   -3.0030
    0.5036
    3.9996
ifail =
```

```
                 0
```