NAG Toolbox for MATLAB

f02ea

1 Purpose

f02ea computes all the eigenvalues, and optionally the Schur factorization, of a real general matrix.

2 Syntax

```
[a, wr, wi, z, ifail] = f02ea(job, a, 'n', n)
```

3 Description

f02ea computes all the eigenvalues, and optionally the Schur form or the complete Schur factorization, of a real general matrix A:

$$A = ZTZ^{\mathrm{T}},$$

where T is an upper quasi-triangular matrix, and Z is an orthogonal matrix. T is called the *Schur form* of A, and the columns of Z are called the *Schur vectors*.

If it is desired to order the Schur factorization so that specified eigenvalues occur in the leading positions on the diagonal of T, then this function may be followed by a call of f08qg. Other reorderings may be achieved by calls to f08qf.

4 References

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

5 Parameters

5.1 Compulsory Input Parameters

1: **job – string**

Indicates whether the Schur form and Schur vectors are to be computed.

job = 'N'

Only eigenvalues are computed.

job = 'S'

Eigenvalues and the Schur form T are computed.

job = 'V'

Eigenvalues, the Schur form and the Schur vectors are computed.

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

2: **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 general matrix A.

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, ldz, 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** = 'S' or 'V', **a** contains the upper quasi-triangular matrix T, the Schur form of A.

If **job** = 'N', the contents of **a** are overwritten.

2: **wr($*$) – double array**

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

The real part of the computed eigenvalues. Complex conjugate pairs of eigenvalues are stored in consecutive elements of **wr** and **wi**, with the eigenvalue having positive imaginary parts first.

If **job** = 'S' or 'V', the eigenvalues occur in the same order as on the diagonal of T, with complex conjugate pairs corresponding to 2 by 2 diagonal blocks.

3: **wi($*$) – double array**

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

The imaginary part of the computed eigenvalues. Complex conjugate pairs of eigenvalues are stored in consecutive elements of **wr** and **wi**, with the eigenvalue having positive imaginary part first.

If **job** = 'S' or 'V', the eigenvalues occur in the same order as on the diagonal of T, with complex conjugate pairs corresponding to 2 by 2 diagonal blocks.

4: **z(ldz,$*$) – double array**

The first dimension, **ldz**, of the array **z** must satisfy

if **job** = 'N' or 'S', $\mathbf{ldz} \geq 1$;
if **job** = 'V', $\mathbf{ldz} \geq \max(1, \mathbf{n})$.

The second dimension of the array must be at least $\max(1, \mathbf{n})$ if **job** = 'V', and at least 1 otherwise

If **job** = 'V', **z** contains the orthogonal matrix Z of Schur vectors.

If **job** = 'N' or 'S', **z** is not referenced.

5: **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', 'S' or 'V',
or **n** < 0,
or **lda** $< \max(1, \mathbf{n})$,
or **ldz** < 1, or **ldz** $<$ **n** and **job** $=$ 'V',
or **lwork** $< \max(1, 3 \times \mathbf{n})$.

ifail $= 2$

The QR algorithm failed to compute all the eigenvalues.

7 Accuracy

The computed Schur factorization is the exact factorization of a nearby matrix $A + E$, where

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

and ϵ is the ***machine precision***.

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

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

where $c(n)$ is a modestly increasing function of n, and s_i is the reciprocal condition number of λ_i. The condition numbers s_i may be computed by calling f08ql.

8 Further Comments

f02ea calls functions from LAPACK in Chapter F08. It first reduces A to upper Hessenberg form H, using an orthogonal similarity transformation: $A = QHQ^{\mathrm{T}}$. If only eigenvalues or the Schur form are required, the function uses the upper Hessenberg QR algorithm to compute the eigenvalues or Schur form of H. If the Schur vectors are required, the function first forms the orthogonal matrix Q that was used in the reduction to Hessenberg form; it then uses the QR algorithm to reduce H to T, using further orthogonal transformations: $H = STS^{\mathrm{T}}$, and at the same time accumulates the matrix of Schur vectors $Z = QS$.

If all the computed eigenvalues are real, then T is upper triangular, and the diagonal elements of T are the eigenvalues of A. If some of the computed eigenvalues form complex conjugate pairs, then T has 2 by 2 diagonal blocks. Each block has the form

$$\begin{pmatrix} \alpha & \beta \\ \gamma & \alpha \end{pmatrix},$$

where $\beta\gamma < 0$, and the corresponding eigenvalues are $\alpha \pm \sqrt{\beta\gamma}$.

Each Schur vector 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';
a = [0.35, 0.45, -0.14, -0.17;
     0.09, 0.07000000000000001, -0.54, 0.35;
     -0.44, -0.33, -0.03, 0.17;
     0.25, -0.32, -0.13, 0.11];
[aOut, wr, wi, z, ifail] = f02ea(job, a)
```

```
aOut =
    0.7995   -0.0060   -0.1144    0.0336
         0   -0.0994    0.6483   -0.2026
         0   -0.2478   -0.0994    0.3474
         0         0         0   -0.1007
wr =
    0.7995
   -0.0994
   -0.0994
   -0.1007
wi =
         0
    0.4008
   -0.4008
         0
z =
    0.6551   -0.3450    0.1037    0.6641
    0.5236    0.6141   -0.5807   -0.1068
   -0.5362    0.2935   -0.3073    0.7293
    0.0956    0.6463    0.7467    0.1249
ifail =
           0
```