# NAG Toolbox for MATLAB

# f02gb

## 1 Purpose

f02gb computes all the eigenvalues, and optionally all the eigenvectors, of a complex general matrix.

## 2 Syntax

```
[a, w, v, ifail] = f02gb(job, a, 'n', n)
```

## 3 Description

f02gb computes all the eigenvalues, and optionally all the right eigenvectors, of a complex general matrix $A$:

$$Ax_i = \lambda_i x_i, \qquad i = 1, 2, \ldots, n.$$

## 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 eigenvectors are to be computed.

**job** = 'N'

Only eigenvalues are computed.

**job** = 'V'

Eigenvalues and eigenvectors are computed.

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

2: **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 $n$ 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, ldv, 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 Schur form of the balanced input matrix $A'$ (see Section 8).

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

2: **w($*$) – complex array**

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

The computed eigenvalues.

3: **v(ldv,$*$) – complex array**

The first dimension, **ldv**, of the array **v** must satisfy

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

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', **v** contains the eigenvectors, with the $i$th column holding the eigenvector associated with the eigenvalue $\lambda_i$ (stored in $\mathbf{w}(i)$).

If **job** $=$ 'N', **v** is not referenced.

4: **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 **n** $< 0$,
or **lda** $< \max(1, \mathbf{n})$,
or **ldv** $< 1$, or **ldv** $< \mathbf{n}$ and **job** $=$ 'V',
or **lwork** $< \max(1, 2 \times \mathbf{n})$.

**ifail** $= 2$

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

## 7 Accuracy

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

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

where $c(n)$ is a modestly increasing function of $n$, $\epsilon$ is the ***machine precision***, and $s_i$ is the reciprocal condition number of $\lambda_i$; $A'$ is the balanced form of the original matrix $A$ (see Section 8), and $\left\|A'\right\| \leq \|A\|$.

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

$$\theta(\tilde{x}_i, x_i) \leq \frac{c(n)\epsilon\|A'\|_2}{sep_i},$$

where $sep_i$ is the reciprocal condition number of $x_i$.

The condition numbers $s_i$ and $sep_i$ may be computed by calling f08qy, using the Schur form of the balanced matrix $A'$ which is returned in the array **a** when **job** = 'V'.

# 8 Further Comments

f02gb calls functions from LAPACK in Chapter F08. It first balances the matrix, using a diagonal similarity transformation to reduce its norm; and then reduces the balanced matrix $A'$ to upper Hessenberg form $H$, using a unitary similarity transformation: $A' = QHQ^{\mathrm{H}}$. If only eigenvalues are required, the function uses the Hessenberg $QR$ algorithm to compute the eigenvalues. If the eigenvectors are required, the function first forms the unitary matrix $Q$ that was used in the reduction to Hessenberg form; it then uses the Hessenberg $QR$ algorithm to compute the Schur factorization of $A'$ as $A' = ZTZ^{\mathrm{H}}$. It computes the right eigenvectors of $T$ by backward substitution, pre-multiplies them by $Z$ to form the eigenvectors of $A'$; and finally transforms the eigenvectors to those of the original matrix $A$.

Each eigenvector $x$ is normalized so that $\|x\|_2 = 1$, 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';
a = [complex(-3.97, -5.04), complex(-4.11, +3.7), complex(-0.34, +1.01),
complex(1.29, -0.86);
       complex(0.34, -1.5), complex(1.52, -0.43), complex(1.88, -5.38),
complex(3.36, +0.65);
       complex(3.31, -3.85), complex(2.5, +3.45), complex(0.88, -1.08),
complex(0.64, -1.48);
      complex(-1.1, +0.82), complex(1.81, -1.59), complex(3.25, +1.33),
complex(1.57, -3.44)];
[aOut, w, v, ifail] = f02gb(job, a)
```

```
aOut =
   -6.0004 - 6.9998i   -0.3656 + 0.3637i    0.4761 - 0.1946i   -0.7237 +
0.5589i
         0             -5.0000 + 2.0060i    0.4981 - 0.5232i   -0.1637 +
0.2071i
         0                  0               7.9982 - 0.9964i    0.8487 -
0.6651i
         0                  0                   0               3.0023 -
3.9998i
w =
  -6.0004 - 6.9998i
  -5.0000 + 2.0060i
   7.9982 - 0.9964i
   3.0023 - 3.9998i
v =
    0.8457             -0.3865 + 0.1732i   -0.1730 + 0.2669i   -0.0356 -
0.1782i
   -0.0177 + 0.3036i   -0.3539 + 0.4529i    0.6924              0.1264 +
0.2666i
    0.0875 + 0.3115i    0.6124              0.3324 + 0.4960i    0.0129 -
0.2966i
  -0.0561 - 0.2906i  -0.0859 - 0.3284i   0.2504 - 0.0147i   0.8898
ifail =
           0
```