NAG Toolbox for MATLAB

f08nb

1 Purpose

f08nb computes the eigenvalues and, optionally, the left and/or right eigenvectors for an n by n real nonsymmetric matrix A.

Optionally, it also computes a balancing transformation to improve the conditioning of the eigenvalues and eigenvectors, reciprocal condition numbers for the eigenvalues, and reciprocal condition numbers for the right eigenvectors.

2 Syntax

```
[a, wr, wi, vl, vr, ilo, ihi, scale, abnrm, rconde, rcondv, info] =
f08nb(balanc, jobvl, jobvr, sense, a, 'n', n)
```

3 Description

The right eigenvector v_j of A satisfies

$$Av_j = \lambda_j v_j$$

where λ_j is the jth eigenvalue of A. The left eigenvector u_j of A satisfies

$$u_j^{\mathrm{H}} A = \lambda_j u_j^{\mathrm{H}}$$

where u_j^{H} denotes the conjugate transpose of u_j.

Balancing a matrix A comprises two operations. The first is to permute the rows and columns of A to make it 'more nearly' upper triangular (closer to Schur form): $A' = PAP^{\mathrm{T}}$, where P is a permutation matrix. The second operation is applying a diagonal similarity transformation DAD^{-1}, where D is a diagonal matrix, with the aim of making its rows and columns closer in norm and the condition numbers of its eigenvalues and eigenvectors smaller. The computed reciprocal condition numbers correspond to the balanced matrix. Permuting rows and columns will not change the condition numbers (in exact arithmetic) but diagonal scaling will. For further explanation of balancing, see Section 4.8.1.2 of Anderson *et al.* 1999.

Following the optional balancing, the matrix A is first reduced to upper Hessenberg form by means of orthogonal similarity transformations, and the QR algorithm is then used to further reduce the matrix to upper quasi-triangular Schur form, T, with 1 by 1 and 2 by 2 blocks on the main diagonal. The eigenvalues are computed from T, the 2 by 2 blocks corresponding to complex conjugate pairs and, optionally, the eigenvectors of T are computed and backtransformed to the eigenvectors of A.

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 URL: 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

5.1 Compulsory Input Parameters

1: **balanc – string**

Indicates how the input matrix should be diagonally scaled and/or permuted to improve the conditioning of its eigenvalues.

balanc = 'N'

Do not diagonally scale or permute.

balanc = 'P'

Perform permutations to make the matrix more nearly upper triangular. Do not diagonally scale.

balanc = 'S'

Diagonally scale the matrix, i.e., replace A by DAD^{-1}, where D is a diagonal matrix chosen to make the rows and columns of A more equal in norm. Do not permute.

balanc = 'B'

Both diagonally scale and permute A.

Computed reciprocal condition numbers will be for the matrix after balancing and/or permuting. Permuting does not change condition numbers (in exact arithmetic), but balancing does.

Constraint: **balanc** = 'N', 'P', 'S' or 'B'.

2: **jobvl – string**

If **jobvl** = 'N', the left eigenvectors of A are not computed.

If **jobvl** = 'V', the left eigenvectors of A are computed.

If **sense** = 'E' or 'B', **jobvl** must be set to **jobvl** = 'V'.

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

3: **jobvr – string**

If **jobvr** = 'N', the right eigenvectors of A are not computed.

If **jobvr** = 'V', the right eigenvectors of **a** are computed.

If **sense** = 'E' or 'B', **jobvr** must be set to **jobvr** = 'V'.

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

4: **sense – string**

Determines which reciprocal condition numbers are computed.

sense = 'N'

None are computed.

sense = 'E'

Computed for eigenvalues only.

sense = 'V'

Computed for right eigenvectors only.

sense = 'B'

Computed for eigenvalues and right eigenvectors.

If **sense** = 'E' or 'B', both left and right eigenvectors must also be computed (**jobvl** = 'V' and **jobvr** = 'V').

Constraint: **sense** = 'N', 'E', 'V' or 'B'.

5: **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 matrix A.

5.2 Optional Input Parameters

1: **n – int32 scalar**

Default: The first dimension of the array **a** and the second dimension of the array **a**. (An error is raised if these dimensions are not equal.)

n, the order of the matrix A.

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

5.3 Input Parameters Omitted from the MATLAB Interface

lda, ldvl, ldvr, work, lwork, iwork

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 $\max(1, \mathbf{n})$

a has been overwritten. If **jobvl** = 'V' or **jobvr** = 'V', A contains the real Schur form of the balanced version of the input matrix A.

2: **wr($*$) – double array**

3: **wi($*$) – double array**

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

wr and **wi** contain the real and imaginary parts, respectively, of the computed eigenvalues. Complex conjugate pairs of eigenvalues appear consecutively with the eigenvalue having the positive imaginary part first.

4: **vl(ldvl,$*$) – double array**

The first dimension, **ldvl**, of the array **vl** must satisfy

if **jobvl** = 'V', **ldvl** $\geq \max(1, \mathbf{n})$;
ldvl ≥ 1 otherwise.

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

If **jobvl** = 'V', the left eigenvectors u_j are stored one after another in the columns of **vl**, in the same order as their corresponding eigenvalues.

If **jobvl** = 'N', **vl** is not referenced.

If the jth eigenvalue is real, then $u_j = \mathbf{vl}(:, j)$, the jth column of **vl**.

If the jth and $(j+1)$st eigenvalues form a complex conjugate pair, then $u_j = \mathbf{vl}(:, j) + i \times \mathbf{vl}(:, j+1)$ and $u_{j+1} = \mathbf{vl}(:, j) - i \times \mathbf{vl}(:, j+1)$.

5: **vr(ldvr,$*$) – double array**

The first dimension, **ldvr**, of the array **vr** must satisfy

if **jobvr** = 'V', **ldvr** $\geq \max(1, \mathbf{n})$;
ldvr ≥ 1 otherwise.

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

If **jobvr** = 'V', the right eigenvectors v_j are stored one after another in the columns of **vr**, in the same order as their corresponding eigenvalues.

If **jobvr** = 'N', **vr** is not referenced.

If the jth eigenvalue is real, then $v_j = \mathbf{vr}(:, j)$, the jth column of **vr**.

If the jth and $(j+1)$st eigenvalues form a complex conjugate pair, then $v_j = \mathbf{vr}(:, j) + i \times \mathbf{vr}(:, j+1)$ and $v_{j+1} = \mathbf{vr}(:, j) - i \times \mathbf{vr}(:, j+1)$.

6: **ilo – int32 scalar**

7: **ihi – int32 scalar**

ilo and **ihi** are integer values determined when A was balanced. The balanced A has $a_{ij} = 0$ if $i > j$ and $j = 1, \ldots, \mathbf{ilo} - 1$ or $i = \mathbf{ihi} + 1, \ldots, \mathbf{n}$.

8: **scale($*$) – double array**

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

Details of the permutations and scaling factors applied when balancing A.

If p_j is the index of the row and column interchanged with row and column j, and d_j is the scaling factor applied to row and column j, then

$\mathbf{scale}(j) = p_j$, for $j = 1, \ldots, \mathbf{ilo} - 1$;

$\mathbf{scale}(j) = d_j$, for $j = \mathbf{ilo}, \ldots, \mathbf{ihi}$;

$\mathbf{scale}(j) = p_j$, for $j = \mathbf{ihi} + 1, \ldots, \mathbf{n}$.

The order in which the interchanges are made is **n** to $\mathbf{ihi} + 1$, then 1 to $\mathbf{ilo} - 1$.

9: **abnrm – double scalar**

The 1-norm of the balanced matrix (the maximum of the sum of absolute values of elements of any column).

10: **rconde**$(*)$ **– double array**

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

rconde(j) is the reciprocal condition number of the jth eigenvalue.

11: **rcondv**$(*)$ **– double array**

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

rcondv(j) is the reciprocal condition number of the jth right eigenvector.

12: **info – int32 scalar**

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

info $= -i$

If **info** $= -i$, parameter i had an illegal value on entry. The parameters are numbered as follows:

1: **balanc**, 2: **jobvl**, 3: **jobvr**, 4: **sense**, 5: **n**, 6: **a**, 7: **lda**, 8: **wr**, 9: **wi**, 10: **vl**, 11: **ldvl**, 12: **vr**, 13: **ldvr**, 14: **ilo**, 15: **ihi**, 16: **scale**, 17: **abnrm**, 18: **rconde**, 19: **rcondv**, 20: **work**, 21: **lwork**, 22: **iwork**, 23: **info**.

It is possible that **info** refers to a parameter that is omitted from the MATLAB interface. This usually indicates that an error in one of the other input parameters has caused an incorrect value to be inferred.

info > 0

If **info** $= i$, the QR algorithm failed to compute all the eigenvalues, and no eigenvectors or condition numbers have been computed; elements $1 : \mathbf{ilo} - 1$ and $i + 1 : \mathbf{n}$ of **wr** and **wi** contain eigenvalues which have converged.

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 4.8 of Anderson *et al.* 1999 for further details.

8 Further Comments

Each eigenvector is normalized to have Euclidean norm equal to unity and the element of largest absolute value real and positive.

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

The complex analogue of this function is f08np.

9 Example

```
balanc = 'Balance';
jobvl = 'Vectors (left)';
jobvr = 'Vectors (right)';
sense = 'Both reciprocal condition numbers';
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, vl, vr, ilo, ihi, scale, abnrm, rconde, rcondv, info] =
f08nb(balanc, jobvl, jobvr, sense, 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
vl =
   -0.6245    0.5330         0    0.6641
   -0.5995   -0.2666    0.4041   -0.1068
    0.4999    0.3455    0.3153    0.7293
   -0.0271   -0.2541   -0.4451    0.1249
vr =
   -0.6551   -0.1933    0.2546    0.1253
   -0.5236    0.2519   -0.5224    0.3320
    0.5362    0.0972   -0.3084    0.5938
   -0.0956    0.6760         0    0.7221
ilo =
           1
ihi =
           4
scale =
     1
     1
     1
     1
abnrm =
    1.1700
rconde =
    0.9936
    0.7027
    0.7027
    0.5710
rcondv =
    0.8181
    0.3996
    0.3996
    0.3125
info =
           0
```