NAG Toolbox

nag_lapack_dgeevx (f08nb)

1 Purpose

nag_lapack_dgeevx (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] =
nag_lapack_dgeevx(balanc, jobvl, jobvr, sense, a, 'n', n)
```

```
[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 means permuting the rows and columns to make it more nearly upper triangular, and 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 unitary similarity transformations, and the QR algorithm is then used to further reduce the matrix to upper triangular Schur form, T, from which the eigenvalues are computed. Optionally, the eigenvectors of T are also computed and backtransformed to those 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 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** – CHARACTER(1)

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** – CHARACTER(1)

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** – CHARACTER(1)

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** – CHARACTER(1)

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**($lda,:$) – REAL (KIND=nag_wp) array

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

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

The n by n matrix A.

5.2 Optional Input Parameters

1: **n** – INTEGER

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 Output Parameters

1: **a**($lda,:$) – REAL (KIND=nag_wp) array

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

The second dimension of the array **a** will be $\max(1,\mathbf{n})$.

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**($:$) – REAL (KIND=nag_wp) array
3: **wi**($:$) – REAL (KIND=nag_wp) array

The dimension of the arrays **wr** and **wi** will be $\max(1,\mathbf{n})$

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**($ldvl,:$) – REAL (KIND=nag_wp) array

The first dimension, $ldvl$, of the array **vl** will be

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

The second dimension of the array **vl** will be $\max(1,\mathbf{n})$ if **jobvl** = 'V' and 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 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 $u_{j+1} = \mathbf{vl}(:,j) - i \times \mathbf{vl}(:,j+1)$.

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

5: **vr**($ldvr,:$) – REAL (KIND=nag_wp) array

The first dimension, $ldvr$, of the array **vr** will be

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

The second dimension of the array **vr** will be $\max(1,\mathbf{n})$ if **jobvr** = 'V' and 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 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 $v_{j+1} = \mathbf{vr}(:,j) - i \times \mathbf{vr}(:,j+1)$.

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

6: **ilo** – INTEGER

7: **ihi** – INTEGER

ilo and **ihi** are integer values determined when A was balanced. The balanced A has $a_{ij} = 0$ if $i > j$ and $j = 1, 2, \ldots,$ **ilo** $- 1$ or $i =$ **ihi** $+ 1, \ldots,$ **n**.

8: **scale**(:) – REAL (KIND=nag_wp) array

The dimension of the array **scale** will be $\max(1,$ **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

scale$(j) = p_j$, for $j = 1, 2, \ldots,$ **ilo** $- 1$;

scale$(j) = d_j$, for $j =$ **ilo**$, \ldots,$ **ihi**;

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

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

9: **abnrm** – REAL (KIND=nag_wp)

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

10: **rconde**(:) – REAL (KIND=nag_wp) array

The dimension of the array **rconde** will be $\max(1,$ **n**$)$

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

11: **rcondv**(:) – REAL (KIND=nag_wp) array

The dimension of the array **rcondv** will be $\max(1,$ **n**$)$

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

12: **info** – INTEGER

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

6 Error Indicators and Warnings

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 (*warning*)

If **info** $= i$, the QR algorithm failed to compute all the eigenvalues, and no eigenvectors or condition numbers have been computed; elements $1 :$ **ilo** $- 1$ and $i + 1 :$ **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.

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

The complex analogue of this function is nag_lapack_zgeevx (f08np).

9 Example

This example finds all the eigenvalues and right eigenvectors of the matrix

$$A = \begin{pmatrix} 0.35 & 0.45 & -0.14 & -0.17 \\ 0.09 & 0.07 & -0.54 & 0.35 \\ -0.44 & -0.33 & -0.03 & 0.17 \\ 0.25 & -0.32 & -0.13 & 0.11 \end{pmatrix},$$

together with estimates of the condition number and forward error bounds for each eigenvalue and eigenvector. The option to balance the matrix is used. In order to compute the condition numbers of the eigenvalues, the left eigenvectors also have to be computed, but they are not printed out in this example.

Note that the block size (NB) of 64 assumed in this example is not realistic for such a small problem, but should be suitable for large problems.

9.1 Program Text

```
     function f08nb_example

fprintf('f08nb example results\n\n');

% Matrix A
n = 4;
a = [0.35,  0.45, -0.14, -0.17;
      0.09,  0.07, -0.54,  0.35;
     -0.44, -0.33, -0.03,  0.17;
      0.25, -0.32, -0.13,  0.11];

% Eigenvalues and left and right eigenvectors of A after matrix balancing
balanc = 'Balance';
jobvl = 'Vectors (left)';
jobvr = 'Vectors (right)';
sense = 'Both reciprocal condition numbers';
[a, wr, wi, vl, vr, ilo, ihi, scale, abnrm, rconde, rcondv, info] = ...
f08nb( ...
       balanc, jobvl, jobvr, sense, a);

disp('Eigenvalues');
fprintf('\n       Eigenvalue                  rcond\n\n');
for j=1:n
  fprintf('%3d',j);
  if wi(j)==0
    fprintf('%15.4e%24.4f\n',wr(j),rconde(j));
  elseif wi(j)<0
    fprintf('%15.4e - %10.4ei%10.4f\n',wr(j),abs(wi(j)),rconde(j));
  else
    fprintf('%15.4e + %10.4ei%10.4f\n',wr(j),wi(j),rconde(j));
  end
```

```
end

fprintf('\nEigenvectors\n\n');
fprintf('       Eigenvector                 rcond\n');
evecs = complex(zeros(n,n));
k = 1;
conjugating = false;
for j = 1:n
  fprintf('\n%3d',j);
  if wi(j)==0 && ~conjugating
    fprintf('%15.4e%24.4f\n',vr(1,k),rcondv(j));
    fprintf('%18.4e\n',vr(2:n,k));
    k = k + 1;
  else
    if conjugating
      pl = '-';
      mi = '+';
    else
      pl = '+';
      mi = '-';
    end
    for l = 1:n
      if (l>1)
        fprintf('%3s', ' ');
      end
      if vr(l,k+1)>0
        fprintf('%15.4e %s %10.4ei', vr(l,k), pl, vr(l,k+1));
      else
        fprintf('%15.4e %s %10.4ei', vr(l,k), mi, abs(vr(l,k+1)));
      end
      if l==1
        fprintf('%10.4f', rcondv(j));
      end
      fprintf('\n');
    end
    if conjugating
      k = k + 2;
    end
    conjugating = ~conjugating;
  end
end
```

9.2 Program Results

```
     f08nb example results

Eigenvalues

        Eigenvalue                 rcond

  1     7.9948e-01                 0.9936
  2    -9.9412e-02 + 4.0079e-01i   0.7027
  3    -9.9412e-02 - 4.0079e-01i   0.7027
  4    -1.0066e-01                 0.5710

Eigenvectors

        Eigenvector                rcond

  1    -6.5509e-01                 0.6252
       -5.2363e-01
        5.3622e-01
       -9.5607e-02

  2    -1.9330e-01 + 2.5463e-01i   0.3996
        2.5186e-01 - 5.2240e-01i
        9.7182e-02 - 3.0838e-01i
        6.7595e-01 - 0.0000e+00i

  3    -1.9330e-01 - 2.5463e-01i   0.3996
```

```
       2.5186e-01 + 5.2240e-01i
       9.7182e-02 + 3.0838e-01i
       6.7595e-01 + 0.0000e+00i

 4     1.2533e-01                  0.3125
       3.3202e-01
       5.9384e-01
       7.2209e-01
```