NAG Toolbox

nag_lapack_dgeev (f08na)

1 Purpose

nag_lapack_dgeev (f08na) computes the eigenvalues and, optionally, the left and/or right eigenvectors for an n by n real nonsymmetric matrix A.

2 Syntax

```
[a, wr, wi, vl, vr, info] = nag_lapack_dgeev(jobvl, jobvr, a, 'n', n)
[a, wr, wi, vl, vr, info] = f08na(jobvl, jobvr, 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.

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 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: **jobvl** – CHARACTER(1)

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

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

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

2: **jobvr** – CHARACTER(1)

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

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

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

3: **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 $\max(1, \mathbf{n})$.

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: **info** – INTEGER

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

6 Error Indicators and Warnings

info < 0

If **info** $= -i$, argument i had an illegal value. An explanatory message is output, and execution of the program is terminated.

info > 0 (*warning*)

The QR algorithm failed to compute all the eigenvalues, and no eigenvectors have been computed; elements $\langle value \rangle$ to **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_zgeev (f08nn).

9 Example

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

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

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 f08na_example


fprintf('f08na 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 right eigenvectors of A
jobvl = 'No left vectors';
jobvr = 'Vectors (right)';
[~, wr, wi, ~, vr, info] = f08na( ...
                                  jobvl, jobvr, a);
```

```
fprintf('Index  Eigenvalue                Eigenvector\n');
k = 1;
conjugating = false;
for j = 1:n
  fprintf('%3d', j);
  if wi(j)==0 & ~conjugating
    fprintf('  %12.4e%15s',wr(j),' ');
    for l = 1:n
      if (l>1)
        fprintf('%32s', ' ');
      end
      fprintf('%12.4e\n',vr(l,k));
    end
    k = k + 1;
  else
    if conjugating
      pl = '-';
      mi = '+';
    else
      pl = '+';
      mi = '-';
    end
    fprintf('  %12.4e %s %10.4ei ', wr(j), pl, abs(wi(j)));
    for l = 1:n
      if (l>1)
        fprintf('%32s', ' ');
      end
      if vr(l,k+1)>0
        fprintf('%12.4e %s %10.4ei\n', vr(l,k), pl, vr(l,k+1));
      else
        fprintf('%12.4e %s %10.4ei\n', vr(l,k), mi, abs(vr(l,k+1)));
      end
    end
    if conjugating
      k = k + 2;
    end
    conjugating = ~conjugating;
  end
  fprintf('\n');
end
```

9.2 Program Results

```
     f08na example results

Index  Eigenvalue                Eigenvector
  1    7.9948e-01                -6.5509e-01
                                 -5.2363e-01
                                  5.3622e-01
                                 -9.5607e-02

  2   -9.9412e-02 + 4.0079e-01i  -1.9330e-01 + 2.5463e-01i
                                  2.5186e-01 - 5.2240e-01i
                                  9.7182e-02 - 3.0838e-01i
                                  6.7595e-01 - 0.0000e+00i

  3   -9.9412e-02 - 4.0079e-01i  -1.9330e-01 - 2.5463e-01i
                                  2.5186e-01 + 5.2240e-01i
                                  9.7182e-02 + 3.0838e-01i
                                  6.7595e-01 + 0.0000e+00i

  4   -1.0066e-01                 1.2533e-01
                                  3.3202e-01
                                  5.9384e-01
                                  7.2209e-01
```