NAG Toolbox

nag_lapack_dhseqr (f08pe)

1 Purpose

nag_lapack_dhseqr (f08pe) computes all the eigenvalues and, optionally, the Schur factorization of a real Hessenberg matrix or a real general matrix which has been reduced to Hessenberg form.

2 Syntax

```
[h, wr, wi, z, info] = nag_lapack_dhseqr(job, compz, ilo, ihi, h, z, 'n', n)
[h, wr, wi, z, info] = f08pe(job, compz, ilo, ihi, h, z, 'n', n)
```

3 Description

nag_lapack_dhseqr (f08pe) computes all the eigenvalues and, optionally, the Schur factorization of a real upper Hessenberg matrix H:

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

where T is an upper quasi-triangular matrix (the Schur form of H), and Z is the orthogonal matrix whose columns are the Schur vectors z_i. See Section 9 for details of the structure of T.

The function may also be used to compute the Schur factorization of a real general matrix A which has been reduced to upper Hessenberg form H:

$$\begin{aligned} A &= QHQ^{\mathrm{T}}, \text{ where } Q \text{ is orthogonal,} \\ &= (QZ)T(QZ)^{\mathrm{T}}. \end{aligned}$$

In this case, after nag_lapack_dgehrd (f08ne) has been called to reduce A to Hessenberg form, nag_lapack_dorghr (f08nf) must be called to form Q explicitly; Q is then passed to nag_lapack_dhseqr (f08pe), which must be called with **compz** = 'V'.

The function can also take advantage of a previous call to nag_lapack_dgebal (f08nh) which may have balanced the original matrix before reducing it to Hessenberg form, so that the Hessenberg matrix H has the structure:

$$\begin{pmatrix} H_{11} & H_{12} & H_{13} \\ & H_{22} & H_{23} \\ & & H_{33} \end{pmatrix}$$

where H_{11} and H_{33} are upper triangular. If so, only the central diagonal block H_{22} (in rows and columns i_{lo} to i_{hi}) needs to be further reduced to Schur form (the blocks H_{12} and H_{23} are also affected). Therefore the values of i_{lo} and i_{hi} can be supplied to nag_lapack_dhseqr (f08pe) directly. Also, nag_lapack_dgebak (f08nj) must be called after this function to permute the Schur vectors of the balanced matrix to those of the original matrix. If nag_lapack_dgebal (f08nh) has not been called however, then i_{lo} must be set to 1 and i_{hi} to n. Note that if the Schur factorization of A is required, nag_lapack_dgebal (f08nh) must **not** be called with **job** = 'S' or 'B', because the balancing transformation is not orthogonal.

nag_lapack_dhseqr (f08pe) uses a multishift form of the upper Hessenberg QR algorithm, due to Bai and Demmel (1989). The Schur vectors are normalized so that $\|z_i\|_2 = 1$, but are determined only to within a factor ± 1.

4 References

Bai Z and Demmel J W (1989) On a block implementation of Hessenberg multishift QR iteration *Internat. J. High Speed Comput.* **1** 97–112

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** – CHARACTER(1)

Indicates whether eigenvalues only or the Schur form T is required.

job = 'E'
Eigenvalues only are required.

job = 'S'
The Schur form T is required.

Constraint: **job** = 'E' or 'S'.

2: **compz** – CHARACTER(1)

Indicates whether the Schur vectors are to be computed.

compz = 'N'
No Schur vectors are computed (and the array **z** is not referenced).

compz = 'V'
The Schur vectors of A are computed (and the array **z** must contain the matrix Q on entry).

compz = 'I'
The Schur vectors of H are computed (and the array **z** is initialized by the function).

Constraint: **compz** = 'N', 'V' or 'I'.

3: **ilo** – INTEGER
4: **ihi** – INTEGER

If the matrix A has been balanced by nag_lapack_dgebal (f08nh), then **ilo** and **ihi** must contain the values returned by that function. Otherwise, **ilo** must be set to 1 and **ihi** to **n**.

Constraint: **ilo** ≥ 1 and $\min(\textbf{ilo}, \textbf{n}) \leq \textbf{ihi} \leq \textbf{n}$.

5: $\textbf{h}(ldh, :)$ – REAL (KIND=nag_wp) array

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

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

The n by n upper Hessenberg matrix H, as returned by nag_lapack_dgehrd (f08ne).

6: $\textbf{z}(ldz, :)$ – REAL (KIND=nag_wp) array

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

if **compz** = 'V' or 'I', $ldz \geq \max(1, \textbf{n})$;
if **compz** = 'N', $ldz \geq 1$.

The second dimension of the array **z** must be at least $\max(1, \textbf{n})$ if **compz** = 'V' or 'I' and at least 1 if **compz** = 'N'.

If **compz** = 'V', **z** must contain the orthogonal matrix Q from the reduction to Hessenberg form.

If **compz** $=$ 'I', **z** need not be set.

5.2 Optional Input Parameters

1: **n** – INTEGER

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

n, the order of the matrix H.

Constraint: **n** ≥ 0.

5.3 Output Parameters

1: **h**$(ldh,:)$ – REAL (KIND=nag_wp) array

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

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

If **job** $=$ 'E', the array contains no useful information.

If **job** $=$ 'S', **h** stores the upper quasi-triangular matrix T from the Schur decomposition (the Schur form) unless **info** > 0.

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

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

The real and imaginary parts, respectively, of the computed eigenvalues, unless **info** > 0 (in which case see Section 6). Complex conjugate pairs of eigenvalues appear consecutively with the eigenvalue having positive imaginary part first. The eigenvalues are stored in the same order as on the diagonal of the Schur form T (if computed); see Section 9 for details.

4: **z**$(ldz,:)$ – REAL (KIND=nag_wp) array

The first dimension, ldz, of the array **z** will be

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

The second dimension of the array **z** will be $\max(1,\mathbf{n})$ if **compz** $=$ 'V' or 'I' and at least 1 if **compz** $=$ 'N'.

If **compz** $=$ 'V' or 'I', **z** contains the orthogonal matrix of the required Schur vectors, unless **info** > 0.

If **compz** $=$ 'N', **z** is not referenced.

5: **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: **job**, 2: **compz**, 3: **n**, 4: **ilo**, 5: **ihi**, 6: **h**, 7: **ldh**, 8: **wr**, 9: **wi**, 10: **z**, 11: **ldz**, 12: **work**, 13: **lwork**, 14: **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*)

The algorithm has failed to find all the eigenvalues after a total of $30 \times (\mathbf{ihi} - \mathbf{ilo} + 1)$ iterations. If $\mathbf{info} = i$, elements $1, 2, \ldots, \mathbf{ilo} - 1$ and $i + 1, i + 2, \ldots, n$ of **wr** and **wi** contain the real and imaginary parts of contain the eigenvalues which have been found.

If **job** = 'E', then on exit, the remaining unconverged eigenvalues are the eigenvalues of the upper Hessenberg matrix $\hat{H}$, formed from $\mathbf{h}(\mathbf{ilo} : \mathbf{info}, \mathbf{ilo} : \mathbf{info})$, i.e., the **ilo** through **info** rows and columns of the final output matrix H.

If **job** = 'S', then on exit

$$(*) \quad H_i U = U\tilde{H}$$

for some matrix U, where H_i is the input upper Hessenberg matrix and $\tilde{H}$ is an upper Hessenberg matrix formed from $\mathbf{h}(\mathbf{info} + 1 : \mathbf{ihi}, \mathbf{info} + 1 : \mathbf{ihi})$.

If **compz** = 'V', then on exit

$$Z_{\text{out}} = Z_{\text{in}} U$$

where U is defined in $(*)$ (regardless of the value of **job**).

If **compz** = 'I', then on exit

$$Z_{\text{out}} = U$$

where U is defined in $(*)$ (regardless of the value of **job**).

If **info** > 0 and **compz** = 'N', then **z** is not accessed.

7 Accuracy

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

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

and ϵ is the ***machine precision***.

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

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

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 nag_lapack_dtrsna (f08ql).

8 Further Comments

The total number of floating-point operations depends on how rapidly the algorithm converges, but is typically about:

$7n^3$ if only eigenvalues are computed;

$10n^3$ if the Schur form is computed;

$20n^3$ if the full Schur factorization is computed.

The Schur form T has the following structure (referred to as **canonical** Schur form).

If all the computed eigenvalues are real, T is upper triangular, and the diagonal elements of T are the eigenvalues; $\mathbf{wr}(i) = t_{ii}$, for $i = 1, 2, \ldots, n$, and $\mathbf{wi}(i) = 0.0$.

If some of the computed eigenvalues form complex conjugate pairs, then T has 2 by 2 diagonal blocks. Each diagonal block has the form

$$\begin{pmatrix} t_{ii} & t_{i,i+1} \\ t_{i+1,i} & t_{i+1,i+1} \end{pmatrix} = \begin{pmatrix} \alpha & \beta \\ \gamma & \alpha \end{pmatrix}$$

where $\beta\gamma < 0$. The corresponding eigenvalues are $\alpha \pm \sqrt{\beta\gamma}$; $\mathbf{wr}(i) = \mathbf{wr}(i+1) = \alpha$; $\mathbf{wi}(i) = +\sqrt{|\beta\gamma|}$; $\mathbf{wi}(i+1) = -\mathbf{wi}(i)$.

The complex analogue of this function is nag_lapack_zhseqr (f08ps).

9 Example

This example computes all the eigenvalues and the Schur factorization of the upper Hessenberg matrix H, where

$$H = \begin{pmatrix} 0.3500 & -0.1160 & -0.3886 & -0.2942 \\ -0.5140 & 0.1225 & 0.1004 & 0.1126 \\ 0.0000 & 0.6443 & -0.1357 & -0.0977 \\ 0.0000 & 0.0000 & 0.4262 & 0.1632 \end{pmatrix}.$$

See also Section 10 in nag_lapack_dorghr (f08nf), which illustrates the use of this function to compute the Schur factorization of a general matrix.

9.1 Program Text

```
function f08pe_example

fprintf('f08pe example results\n\n');

% Matrix A
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];

% Reduce A to upper Hessenberg Form A = QHQ^T
ilo = nag_int(1);
ihi = nag_int(4);
[H, tau, info] = f08ne(ilo, ihi, a);

% Form Q
[Q, info] = f08nf(ilo, ihi, H, tau);

% Schur factorize H = Y*T*Y' and form Z = QY  A = QY*T*(QQY)'
job   = 'Schur form';
compz = 'Vectors';
[~, wr, wi, Z, info] = f08pe( ...
                              job, compz, ilo, ihi, H, Q);

w = wr + i*wi;
disp('Eigenvalues of A');
disp(w);
```

9.2 Program Results

```
f08pe example results

Eigenvalues of A
   0.7995 + 0.0000i
  -0.0994 + 0.4008i
  -0.0994 - 0.4008i
  -0.1007 + 0.0000i
```