NAG Toolbox

nag_lapack_zhptrd (f08gs)

1 Purpose

nag_lapack_zhptrd (f08gs) reduces a complex Hermitian matrix to tridiagonal form, using packed storage.

2 Syntax

```
[ap, d, e, tau, info] = nag_lapack_zhptrd(uplo, n, ap)
[ap, d, e, tau, info] = f08gs(uplo, n, ap)
```

3 Description

nag_lapack_zhptrd (f08gs) reduces a complex Hermitian matrix A, held in packed storage, to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$.

The matrix Q is not formed explicitly but is represented as a product of $n-1$ elementary reflectors (see the F08 Chapter Introduction for details). Functions are provided to work with Q in this representation (see Section 9).

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: **uplo** – CHARACTER(1)

Indicates whether the upper or lower triangular part of A is stored.

uplo = 'U'
The upper triangular part of A is stored.

uplo = 'L'
The lower triangular part of A is stored.

Constraint: **uplo** = 'U' or 'L'.

2: **n** – INTEGER

n, the order of the matrix A.

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

3: **ap**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **ap** must be at least $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$

The upper or lower triangle of the n by n Hermitian matrix A, packed by columns.

More precisely,

if **uplo** = 'U', the upper triangle of A must be stored with element A_{ij} in **ap**$(i + j(j-1)/2)$ for $i \leq j$;

if **uplo** = 'L', the lower triangle of A must be stored with element A_{ij} in **ap**$(i + (2n-j)(j-1)/2)$ for $i \geq j$.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **ap**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **ap** will be $\max(1, \mathbf{n} \times (\mathbf{n}+1)/2)$

ap stores the tridiagonal matrix T and details of the unitary matrix Q.

2: **d**(**n**) – REAL (KIND=nag_wp) array

The diagonal elements of the tridiagonal matrix T.

3: **e**(**n** − **1**) – REAL (KIND=nag_wp) array

The off-diagonal elements of the tridiagonal matrix T.

4: **tau**(**n** − **1**) – COMPLEX (KIND=nag_wp) array

Further details of the unitary matrix Q.

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: **uplo**, 2: **n**, 3: **ap**, 4: **d**, 5: **e**, 6: **tau**, 7: **info**.

7 Accuracy

The computed tridiagonal matrix T is exactly similar to a nearby matrix $(A + E)$, where

$$\|E\|_2 \leq c(n)\epsilon\|A\|_2,$$

$c(n)$ is a modestly increasing function of n, and ϵ is the ***machine precision***.

The elements of T themselves may be sensitive to small perturbations in A or to rounding errors in the computation, but this does not affect the stability of the eigenvalues and eigenvectors.

8 Further Comments

The total number of real floating-point operations is approximately $\frac{16}{3}n^3$.

To form the unitary matrix Q nag_lapack_zhptrd (f08gs) may be followed by a call to nag_lapack_zupgtr (f08gt):

```
[q, info] = f08gt(uplo, n, ap, tau);
```

To apply Q to an n by p complex matrix C nag_lapack_zhptrd (f08gs) may be followed by a call to nag_lapack_zupmtr (f08gu). For example,

```
[ap, c, info] = f08gu('Left', uplo, 'No Transpose', ap, tau, c);
```

forms the matrix product QC.

The real analogue of this function is nag_lapack_dsptrd (f08ge).

9 Example

This example reduces the matrix A to tridiagonal form, where

$$A = \begin{pmatrix} -2.28+0.00i & 1.78-2.03i & 2.26+0.10i & -0.12+2.53i \\ 1.78+2.03i & -1.12+0.00i & 0.01+0.43i & -1.07+0.86i \\ 2.26-0.10i & 0.01-0.43i & -0.37+0.00i & 2.31-0.92i \\ -0.12-2.53i & -1.07-0.86i & 2.31+0.92i & -0.73+0.00i \end{pmatrix},$$

using packed storage.

9.1 Program Text

```
function f08gs_example

fprintf('f08gs example results\n\n');

% Hermitian matrix A stored in symmetric packed format (Lower)
uplo = 'L';
n = nag_int(4);
ap = [-2.28 + 0i;   1.78 + 2.03i;   2.26 - 0.10i;  -0.12 - 2.53i;
                   -1.12 + 0i;      0.01 - 0.43i;  -1.07 - 0.86i;
                                   -0.37 + 0i;      2.31 + 0.92i;
                                   -0.73 + 0i];

% Reduce to tridiagonal form
[apf, d, e, tau, info] = f08gs( ...
                                uplo, n, ap);

% Note: absolute values for e are displayed because the signs may change
%       with changes in sign of columns of Q.
fprintf('Diagonal and off-diagonal elements of tridiagonal form\n\n');
fprintf('    i         d           e\n');

for j = 1:n-1
  fprintf('%5d%12.5f%12.5f\n', j, d(j), abs(e(j)));
end
fprintf('%5d%12.5f\n', n, d(n));
```

9.2 Program Results

```
f08gs example results

Diagonal and off-diagonal elements of tridiagonal form

    i         d           e
    1    -2.28000     4.33846
    2    -0.12846     2.02259
    3    -0.16659     1.80232
    4    -1.92495
```