NAG Toolbox

nag_lapack_zungtr (f08ft)

1 Purpose

nag_lapack_zungtr (f08ft) generates the complex unitary matrix Q, which was determined by nag_lapack_zhetrd (f08fs) when reducing a Hermitian matrix to tridiagonal form.

2 Syntax

```
[a, info] = nag_lapack_zungtr(uplo, a, tau, 'n', n)
[a, info] = f08ft(uplo, a, tau, 'n', n)
```

3 Description

nag_lapack_zungtr (f08ft) is intended to be used after a call to nag_lapack_zhetrd (f08fs), which reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. nag_lapack_zhetrd (f08fs) represents the unitary matrix Q as a product of $n - 1$ elementary reflectors.

This function may be used to generate Q explicitly as a square matrix.

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)

This **must** be the same argument **uplo** as supplied to nag_lapack_zhetrd (f08fs).

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

2: $\mathbf{a}(lda,:)$ – COMPLEX (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 $\max(1,\mathbf{n})$.

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_zhetrd (f08fs).

3: $\mathbf{tau}(:)$ – COMPLEX (KIND=nag_wp) array

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

Further details of the elementary reflectors, as returned by nag_lapack_zhetrd (f08fs).

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the first dimension of the array **a** and the second dimension of the array **a**.

n, the order of the matrix Q.

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

5.3 Output Parameters

1: $\mathbf{a}(lda,:)$ – COMPLEX (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 $\max(1,\mathbf{n})$.

The n by n unitary matrix Q.

2: **info** – INTEGER

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

6 Error Indicators and Warnings

$\mathbf{info} = -i$

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

1: **uplo**, 2: **n**, 3: **a**, 4: **lda**, 5: **tau**, 6: **work**, 7: **lwork**, 8: **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.

7 Accuracy

The computed matrix Q differs from an exactly unitary matrix by a matrix E such that

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

where ϵ is the ***machine precision***.

8 Further Comments

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

The real analogue of this function is nag_lapack_dorgtr (f08ff).

9 Example

This example computes all the eigenvalues and eigenvectors of the matrix A, where

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

Here A is Hermitian and must first be reduced to tridiagonal form by nag_lapack_zhetrd (f08fs). The program then calls nag_lapack_zungtr (f08ft) to form Q, and passes this matrix to nag_lapack_zsteqr (f08js) which computes the eigenvalues and eigenvectors of A.

9.1 Program Text

```
function f08ft_example

fprintf('f08ft example results\n\n');

% Eigenvalues / vectors of Hermitian matrix A
uplo = 'L';
n = 4;
a = [-2.28 + 0.00i,  0.00 + 0i,     0    + 0i,     0    + 0i;
      1.78 + 2.03i, -1.12 + 0i,     0    + 0i,     0    + 0i;
      2.26 - 0.10i,  0.01 - 0.43i, -0.37 + 0i,     0    + 0i;
     -0.12 - 2.53i, -1.07 - 0.86i,  2.31 + 0.92i, -0.73 + 0i];

% A --> QTQ^H, for tridiagonal T
[QT, d, e, tau, info] = f08fs( ...
                               uplo, a);

% Form Q
[Q, info] = f08ft(uplo, QT, tau);

% Calculate eigenvalues/vectors of A from Q, d and e.
compz = 'V';
[w, ~, z, info] = f08js( ...
                         compz, d, e, Q);

% Normalize vectors, largest element is real and positive.
for i = 1:n
  [~,k] = max(abs(real(z(:,i)))+abs(imag(z(:,i))));
  z(:,i) = z(:,i)*conj(z(k,i))/abs(z(k,i));
end

disp(' Eigenvalues of A:');
disp(w);
disp(' Corresponding eigenvectors:');
disp(z);
```

9.2 Program Results

```
     f08ft example results

 Eigenvalues of A:
   -6.0002
   -3.0030
    0.5036
    3.9996

 Corresponding eigenvectors:
   0.7299 + 0.0000i  -0.2120 + 0.1497i   0.1000 - 0.3570i   0.1991 + 0.4720i
  -0.1663 - 0.2061i   0.7307 + 0.0000i   0.2863 - 0.3353i  -0.2467 + 0.3751i
  -0.4165 - 0.1417i  -0.3291 + 0.0479i   0.6890 + 0.0000i   0.4468 + 0.1466i
   0.1743 + 0.4162i   0.5200 + 0.1329i   0.0662 + 0.4347i   0.5612 + 0.0000i
```