NAG Toolbox

nag_lapack_zupmtr (f08gu)

1 Purpose

nag_lapack_zupmtr (f08gu) multiplies an arbitrary complex matrix C by the complex unitary matrix Q which was determined by nag_lapack_zhptrd (f08gs) when reducing a complex Hermitian matrix to tridiagonal form.

2 Syntax

```
[ap, c, info] = nag_lapack_zupmtr(side, uplo, trans, ap, tau, c, 'm', m, 'n', n)
[ap, c, info] = f08gu(side, uplo, trans, ap, tau, c, 'm', m, 'n', n)
```

3 Description

nag_lapack_zupmtr (f08gu) is intended to be used after a call to nag_lapack_zhptrd (f08gs), which reduces a complex Hermitian matrix A to real symmetric tridiagonal form T by a unitary similarity transformation: $A = QTQ^{\mathrm{H}}$. nag_lapack_zhptrd (f08gs) represents the unitary matrix Q as a product of elementary reflectors.

This function may be used to form one of the matrix products

$$QC, Q^{\mathrm{H}}C, CQ \text{ or } CQ^{\mathrm{H}},$$

overwriting the result on C (which may be any complex rectangular matrix).

A common application of this function is to transform a matrix Z of eigenvectors of T to the matrix QZ of eigenvectors of A.

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: **side** – CHARACTER(1)

Indicates how Q or Q^{H} is to be applied to C.

side = 'L'
: Q or Q^{H} is applied to C from the left.

side = 'R'
: Q or Q^{H} is applied to C from the right.

Constraint: **side** = 'L' or 'R'.

2: **uplo** – CHARACTER(1)

This **must** be the same argument **uplo** as supplied to nag_lapack_zhptrd (f08gs).

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

3: **trans** – CHARACTER(1)

Indicates whether Q or Q^{H} is to be applied to C.

trans = 'N'
Q is applied to C.

trans = 'C'
Q^{H} is applied to C.

Constraint: **trans** = 'N' or 'C'.

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

The dimension of the array **ap** must be at least $\max(1, \mathbf{m} \times (\mathbf{m} + 1)/2)$ if **side** = 'L' and at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$ if **side** = 'R'

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_zhptrd (f08gs).

5: **tau**(:) – COMPLEX (KIND=nag_wp) array

The dimension of the array **tau** must be at least $\max(1, \mathbf{m} - 1)$ if **side** = 'L' and at least $\max(1, \mathbf{n} - 1)$ if **side** = 'R'

Further details of the elementary reflectors, as returned by nag_lapack_zhptrd (f08gs).

6: **c**(ldc, :) – COMPLEX (KIND=nag_wp) array

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

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

The m by n matrix C.

5.2 Optional Input Parameters

1: **m** – INTEGER

Default: the first dimension of the array **c**.

m, the number of rows of the matrix C; m is also the order of Q if **side** = 'L'.

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

2: **n** – INTEGER

Default: the second dimension of the array **c**.

n, the number of columns of the matrix C; n is also the order of Q if **side** = 'R'.

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

5.3 Output Parameters

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

The dimension of the array **ap** will be $\max(1, \mathbf{m} \times (\mathbf{m} + 1)/2)$ if **side** = 'L' and at least $\max(1, \mathbf{n} \times (\mathbf{n} + 1)/2)$ if **side** = 'R'

Is used as internal workspace prior to being restored and hence is unchanged.

2: **c**(ldc, :) – COMPLEX (KIND=nag_wp) array

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

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

c stores QC or $Q^{\mathrm{H}}C$ or CQ or CQ^{H} as specified by **side** and **trans**.

3: **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: **side**, 2: **uplo**, 3: **trans**, 4: **m**, 5: **n**, 6: **ap**, 7: **tau**, 8: **c**, 9: **ldc**, 10: **work**, 11: **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 result differs from the exact result by a matrix E such that

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

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

8 Further Comments

The total number of real floating-point operations is approximately $8m^2n$ if **side** $=$ 'L' and $8mn^2$ if **side** $=$ 'R'.

The real analogue of this function is nag_lapack_dopmtr (f08gg).

9 Example

This example computes the two smallest eigenvalues, and the associated eigenvectors, of the matrix A, where

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

using packed storage. Here A is Hermitian and must first be reduced to tridiagonal form T by nag_lapack_zhptrd (f08gs). The program then calls nag_lapack_dstebz (f08jj) to compute the requested eigenvalues and nag_lapack_zstein (f08jx) to compute the associated eigenvectors of T. Finally nag_lapack_zupmtr (f08gu) is called to transform the eigenvectors to those of A.

9.1 Program Text

```
     function f08gu_example

fprintf('f08gu 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);
```

```
% Calculate two smallest eigenvalues
range = 'Indices';
order = 'Block';
vl = 0;
vu = 0;
il = nag_int(1);
iu = nag_int(2);
abstol = 0;
[m, nsplit, w, iblock, isplit, info] = ...
  f08jj( ...
         range, order, vl, vu, il, iu, abstol, d, e);

% Corresponding eigenvectors of T
[tz, ifailv, info] = f08jx( ...
                            d, e, m, w, iblock, isplit);

% Transform to eigenvalues of A (by premultiplying by Q)
side = 'Left';
trans = 'No transpose';
[~, z, info] = f08gu( ...
                      side, uplo, trans, apf, tau, tz);

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

disp(' Selected eigenvalues of A:');
disp(w(1:m));
disp(' Corresponding eigenvectors:');
disp(z);
```

9.2 Program Results

```
     f08gu example results

 Selected eigenvalues of A:
   -6.0002
   -3.0030

 Corresponding eigenvectors:
   0.7299 + 0.0000i  -0.2120 + 0.1497i
  -0.1663 - 0.2061i   0.7307 + 0.0000i
  -0.4165 - 0.1417i  -0.3291 + 0.0479i
   0.1743 + 0.4162i   0.5200 + 0.1329i
```