NAG Toolbox

nag_lapack_zunmqr (f08au)

1 Purpose

nag_lapack_zunmqr (f08au) multiplies an arbitrary complex matrix C by the complex unitary matrix Q from a QR factorization computed by nag_lapack_zgeqrf (f08as), nag_lapack_zgeqpf (f08bs) or nag_lapack_zgeqp3 (f08bt).

2 Syntax

```
[c, info] = nag_lapack_zunmqr(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
[c, info] = f08au(side, trans, a, tau, c, 'm', m, 'n', n, 'k', k)
```

3 Description

nag_lapack_zunmqr (f08au) is intended to be used after a call to nag_lapack_zgeqrf (f08as), nag_lapack_zgeqpf (f08bs) or nag_lapack_zgeqp3 (f08bt), which perform a QR factorization of a complex matrix A. The unitary matrix Q is represented 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 in solving linear least squares problems, as described in the F08 Chapter Introduction and illustrated in Section 10 in nag_lapack_zgeqrf (f08as).

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: **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'.

3: **a**$(lda,:)$ – COMPLEX (KIND=nag_wp) array

The first dimension, lda, of the array **a** must satisfy

if **side** = 'L', $lda \geq \max(1,\mathbf{m})$;
if **side** = 'R', $lda \geq \max(1,\mathbf{n})$.

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

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_zgeqrf (f08as), nag_lapack_zgeqpf (f08bs) or nag_lapack_zgeqp3 (f08bt).

4: **tau**$(:)$ – COMPLEX (KIND=nag_wp) array

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

Further details of the elementary reflectors, as returned by nag_lapack_zgeqrf (f08as), nag_lapack_zgeqpf (f08bs) or nag_lapack_zgeqp3 (f08bt).

5: **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.

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.

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

3: **k** – INTEGER

Default: the second dimension of the arrays **a**, **tau**.

k, the number of elementary reflectors whose product defines the matrix Q.

Constraints:

if **side** = 'L', $\mathbf{m} \geq \mathbf{k} \geq 0$;
if **side** = 'R', $\mathbf{n} \geq \mathbf{k} \geq 0$.

5.3 Output Parameters

1: **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**.

2: **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: **trans**, 3: **m**, 4: **n**, 5: **k**, 6: **a**, 7: **lda**, 8: **tau**, 9: **c**, 10: **ldc**, 11: **work**, 12: **lwork**, 13: **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 $8nk(2m-k)$ if **side** = 'L' and $8mk(2n-k)$ if **side** = 'R'.

The real analogue of this function is nag_lapack_dormqr (f08ag).

9 Example

See Section 10 in nag_lapack_zgeqrf (f08as).

9.1 Program Text

```
function f08au_example

fprintf('f08au example results\n\n');

a = [ 0.96 - 0.81i, -0.03 + 0.96i, -0.91 + 2.06i, -0.05 + 0.41i;
     -0.98 + 1.98i, -1.20 + 0.19i, -0.66 + 0.42i, -0.81 + 0.56i;
      0.62 - 0.46i,  1.01 + 0.02i,  0.63 - 0.17i, -1.11 + 0.60i;
     -0.37 + 0.38i,  0.19 - 0.54i, -0.98 - 0.36i,  0.22 - 0.20i;
      0.83 + 0.51i,  0.20 + 0.01i, -0.17 - 0.46i,  1.47 + 1.59i;
      1.08 - 0.28i,  0.20 - 0.12i, -0.07 + 1.23i,  0.26 + 0.26i];

% Compute the QR factorization of A
[qr, tau, info] = f08as(a);

c = [-2.09 + 1.93i,  3.26 - 2.70i;
      3.34 - 3.53i, -6.22 + 1.16i;
     -4.94 - 2.04i,  7.94 - 3.13i;
      0.17 + 4.23i,  1.04 - 4.26i;
     -5.19 + 3.63i, -2.31 - 2.12i;
      0.98 + 2.53i, -1.39 - 4.05i];

% Perform B = Q'H*C
side = 'Left';
trans = 'Conjugate transpose';
```

```
[b, info] = f08au( ...
    side, trans, qr, tau, c);

disp('B = Q^H*C:');
disp(b);
```

9.2 Program Results

```
     f08au example results

B = Q^H*C:
   5.3510 - 0.1638i  -4.7626 - 2.8427i
  -5.7559 - 0.2004i   6.3325 - 2.7406i
  -2.5366 + 4.0215i   6.4835 - 3.8629i
  -1.0677 - 6.3316i   6.4968 - 0.7809i
  -0.0381 - 0.0273i  -0.1320 - 0.0612i
  -0.0144 + 0.0483i   0.0906 - 0.0740i
```