

NAG Toolbox

nag_lapack_dormtr (f08fg)

1 Purpose

nag_lapack_dormtr (f08fg) multiplies an arbitrary real matrix C by the real orthogonal matrix Q which was determined by nag_lapack_dsytrd (f08fe) when reducing a real symmetric matrix to tridiagonal form.

2 Syntax

```
[c, info] = nag_lapack_dormtr(side, uplo, trans, a, tau, c, 'm', m, 'n', n)
[c, info] = f08fg(side, uplo, trans, a, tau, c, 'm', m, 'n', n)
```

3 Description

nag_lapack_dormtr (f08fg) is intended to be used after a call to nag_lapack_dsytrd (f08fe), which reduces a real symmetric matrix A to symmetric tridiagonal form T by an orthogonal similarity transformation: $A = QTQ^T$. nag_lapack_dsytrd (f08fe) represents the orthogonal matrix Q as a product of elementary reflectors.

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

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

overwriting the result on C (which may be any real 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^T is to be applied to C .

side = 'L'

Q or Q^T is applied to C from the left.

side = 'R'

Q or Q^T 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_dsytrd (f08fe).

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

3: **trans** – CHARACTER(1)

Indicates whether Q or Q^T is to be applied to C .

trans = 'N'

Q is applied to C .

trans = 'T'

Q^T is applied to C .

Constraint: **trans** = 'N' or 'T'.

4: **a**(*lda*,:) – REAL (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 $\max(1, \mathbf{m})$ if **side** = 'L' and at least $\max(1, \mathbf{n})$ if **side** = 'R'.

Details of the vectors which define the elementary reflectors, as returned by nag_lapack_dsytrd (f08fe).

5: **tau**(:) – REAL (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_dsytrd (f08fe).

6: **c**(*ldc*,:) – REAL (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: **c**(*ldc*,:) – REAL (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^TC or CQ or CQ^T 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: **uplo**, 3: **trans**, 4: **m**, 5: **n**, 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 floating-point operations is approximately $2m^2n$ if **side** = 'L' and $2mn^2$ if **side** = 'R'.

The complex analogue of this function is nag_lapack_zunmtr (f08fu).

9 Example

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

$$A = \begin{pmatrix} 2.07 & 3.87 & 4.20 & -1.15 \\ 3.87 & -0.21 & 1.87 & 0.63 \\ 4.20 & 1.87 & 1.15 & 2.06 \\ -1.15 & 0.63 & 2.06 & -1.81 \end{pmatrix}.$$

Here A is symmetric and must first be reduced to tridiagonal form T by nag_lapack_dsytrd (f08fe). The program then calls nag_lapack_dstebz (f08jj) to compute the requested eigenvalues and nag_lapack_dstein (f08jk) to compute the associated eigenvectors of T . Finally nag_lapack_dormtr (f08fg) is called to transform the eigenvectors to those of A .

9.1 Program Text

```
function f08fg_example

fprintf('f08fg example results\n\n');

% Find eigenvalues 1:2 of A
a = [ 2.07, 0, 0, 0;
      3.87, -0.21, 0, 0;
      4.20, 1.87, 1.15, 0;
      -1.15, 0.63, 2.06, -1.81];

% A --> Q T Q^T, for tridiagonal T with diagonal Td and off-diagonal Tod.
uplo = 'L';
[QT, Td, Tod, tau, info] = f08fe( ...
    uplo, a);
```

```

% Get eigenvalues 1:2 of T (= eigenvalues of A)
vl = 0;
vu = 0;
il = nag_int(1);
iu = nag_int(2);
abstol = 0;
[m, ~, w, iblock, isplit, info] = ...
    f08jj(...
        'I', 'B', vl, vu, il, iu, abstol, Td, Tod);

% Get corresponding eigenvectors of T
[v, ifailv, info] = f08jk( ...
    Td, Tod, m, w, iblock, isplit);

% Transform Q*V to get eigenvectors of A
side = 'Left';
trans = 'No transpose';
[z, info] = f08fg( ...
    side, uplo, trans, QT, tau, v);

fprintf(' Eigenvalues numbered 1 to 2 are:\n  ');
fprintf(' %7.4f',w(1:m));
fprintf('\n\n');

[ifail] = x04ca( ...
    'General', ' ', z, 'Corresponding eigenvectors of A');

```

9.2 Program Results

f08fg example results

Eigenvalues numbered 1 to 2 are:
-5.0034 -1.9987

Corresponding eigenvectors of A

	1	2
1	0.5658	-0.2328
2	-0.3478	0.7994
3	-0.4740	-0.4087
4	0.5781	0.3737
