

nag_real_qr (f01qcc)

1. Purpose

nag_real_qr (f01qcc) finds the QR factorization of the real m by n matrix A , where $m \geq n$.

2. Specification

```
#include <nag.h>
#include <nagf01.h>

void nag_real_qr(Integer m, Integer n, double a[], Integer tda,
                 double zeta[], NagError *fail)
```

3. Description

The m by n matrix A is factorized as

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} \quad \text{when } m > n,$$

$$A = QR \quad \text{when } m = n,$$

where Q is an m by m orthogonal matrix and R is an n by n upper triangular matrix. The factorization is obtained by Householder's method. The k th transformation matrix, Q_k , which is used to introduce zeros into the k th column of A is given in the form

$$Q_k = \begin{pmatrix} I & 0 \\ 0 & T_k \end{pmatrix}$$

where

$$T_k = I - u_k u_k^T,$$

$$u_k = \begin{pmatrix} \zeta_k \\ z_k \end{pmatrix},$$

ζ_k is a scalar and z_k is an $(m - k)$ element vector. ζ_k and z_k are chosen to annihilate the elements below the triangular part of A .

The vector u_k is returned in the $(k - 1)$ th element of the array **zeta** and in the $(k - 1)$ th column of **a**, such that ζ_k is in **zeta**[$k - 1$] and the elements of z_k are in **a**[k][$k - 1$], ..., **a**[$m - 1$][$k - 1$]. The elements of R are returned in the upper triangular part of **a**. Q is given by

$$Q = (Q_n Q_{n-1} \dots Q_1)^T.$$

Good background descriptions to the QR factorization are given in Dongarra *et al*(1979) and Golub and Van Loan (1989).

4. Parameters

m

Input: m , the number of rows of A .

Constraint: **m** $\geq$ **n**.

n

Input: n , the number of columns of A .

When **n** = 0 then an immediate return is effected.

Constraint: **n** $\geq$ 0.

a[**m**][**tda**]

Input: the leading m by n part of the array **a** must contain the matrix to be factorized.

Output: the n by n upper triangular part of **a** will contain the upper triangular matrix R and the m by n strictly lower triangular part of **a** will contain details of the factorization as described in Section 3.

tda

Input: the second dimension of the array **a** as declared in the function from which nag_real_qr is called.

Constraint: **tda** $\geq$ **n**.

zeta[n]

Output: **zeta** [$k - 1$] contains the scalar ζ_k for the k th transformation. If $T_k = I$ then **zeta**($k - 1$) = 0.0, otherwise **zeta** [$k - 1$] contains ζ_k as described in Section 3 and ζ_k is always in the range $(1.0, \sqrt{2.0})$.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_2_INT_ARG_LT

On entry, **m** = $\langle value \rangle$ while **n** = $\langle value \rangle$. These parameters must satisfy **m** $\geq$ **n**.

On entry, **tda** = $\langle value \rangle$ while **n** = $\langle value \rangle$. These parameters must satisfy **tda** $\geq$ **n**.

NE_INT_ARG_LT

On entry, **n** must not be less than 0: **n** = $\langle value \rangle$.

6. Further Comments

The approximate number of floating-point operations is given by $2n^2(3m - n)/3$.

6.1. Accuracy

The computed factors Q and R satisfy the relation

$$Q \begin{pmatrix} R \\ 0 \end{pmatrix} = A + E$$

where $\|E\| \leq c\epsilon\|A\|$, and ϵ is the **machine precision**, c is a modest function of m and n and $\|\cdot\|$ denotes the spectral (two) norm.

6.2. References

Dongarra J J, Moler C B, Bunch J R and Stewart G W (1979) *LINPACK Users' Guide* SIAM, Philadelphia.

Golub G H and Van Loan C F (1989) *Matrix Computations* (2nd Edn) Johns Hopkins University Press, Baltimore.

Wilkinson J H (1965) *The Algebraic Eigenvalue Problem* Clarendon Press, Oxford.

7. See Also

nag_real_apply_q (f01qdc)

nag_real_form_q (f01qec)

8. Example

To obtain the QR factorization of the 5 by 3 matrix

$$A = \begin{pmatrix} 2.0 & 2.5 & 2.5 \\ 2.0 & 2.5 & 2.5 \\ 1.6 & -0.4 & 2.8 \\ 2.0 & -0.5 & 0.5 \\ 1.2 & -0.3 & -2.9 \end{pmatrix}.$$

8.1. Program Text

```

/* nag_real_qr(f01qcc) Example Program
*
* Copyright 1990 Numerical Algorithms Group.
*
* Mark 1, 1990.
*/

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagf01.h>

#define MMAX 20
#define NMAX 10

main()
{
    Integer tda = NMAX;

    double zeta[NMAX], a[MMAX][NMAX];
    Integer i, j, m, n;

    Vprintf("f01qcc Example Program Results\n");
    Vscanf(" %*[^\\n]"); /* skip headings in data file */
    Vscanf(" %*[^\\n]");

    Vscanf("%ld%ld", &m, &n);
    if (m > MMAX || n > NMAX)
    {
        Vprintf("m or n is out of range.\n");
        Vprintf("m = %2ld, n = %2ld\n", m, n);
    }
    else
    {
        Vscanf(" %*[^\\n]"); /* skip next heading */
        for (i = 0; i < m; ++i) /* Read matrix A */
            for (j = 0; j < n; ++j)
                Vscanf("%lf", &a[i][j]);

        /* Find the QR factorization of A */
        f01qcc(m, n, (double *)a, tda, zeta, NAGERR_DEFAULT);

        Vprintf("QR factorization of A\n\n");
        Vprintf("Vector zeta\n");
        for (i = 0; i < n; ++i)
            Vprintf(" %8.4f", zeta[i]);
        Vprintf("\n\n");
        Vprintf("Matrix A after factorization (upper triangular part is R)\n");
        for (i = 0; i < m; ++i)
        {
            for (j = 0; j < n; ++j)
                Vprintf(" %8.4f", a[i][j]);
            Vprintf("\n");
        }
        exit(EXIT_SUCCESS);
    }
}

```

8.2. Program Data

f01qcc Example Program Data
Values of m and n.
5 3
Matrix A
2.0 2.5 2.5
2.0 2.5 2.5
1.6 -0.4 2.8
2.0 -0.5 0.5
1.2 -0.3 -2.9

8.3. Program Results

f01qcc Example Program Results
QR factorization of A

Vector zeta

1.2247 1.1547 1.2649

Matrix A after factorization (upper triangular part is R)

-4.0000	-2.0000	-3.0000
0.4082	-3.0000	-2.0000
0.3266	-0.4619	-4.0000
0.4082	-0.5774	0.0000
0.2449	-0.3464	-0.6325
