NAG Library Function Document

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 kth transformation matrix, Q_k, which is used to introduce zeros into the kth column of A is given in the form

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

where

$$T_k = I - u_k u_k^{\mathrm{T}},$$

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

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 $\mathbf{zeta}[k-1]$ and the elements of z_k are in $\mathbf{a}[(k) \times \mathbf{tda} + k - 1], \ldots, \mathbf{a}[(m-1) \times \mathbf{tda} + k - 1]$. The elements of R are returned in the upper triangular part of **a**. Q is given by

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

Good background descriptions to the QR factorization are given in Dongarra *et al.* (1979) and Golub and Van Loan (1996).

4 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 (1996) *Matrix Computations* (3rd Edition) Johns Hopkins University Press, Baltimore

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

5 Arguments

1: **m** – Integer *Input*

On entry: m, the number of rows of A.

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

2: **n** – Integer *Input*

On entry: n, the number of columns of A.

When $\mathbf{n} = 0$ then an immediate return is effected.

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

3: **a**[**m** × **tda**] – double *Input/Output*

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

On exit: 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

4: **tda** – Integer *Input*

On entry: the stride separating matrix column elements in the array **a**.

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

5: **zeta**[**n**] – double *Output*

On exit: **zeta**$[k-1]$ contains the scalar ζ_k for the kth 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 $\left(1.0, \sqrt{2.0}\right)$.

6: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_2_INT_ARG_LT

On entry, $\mathbf{m} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$. These arguments must satisfy $\mathbf{m} \geq \mathbf{n}$.

On entry, $\mathbf{tda} = \langle value \rangle$ while $\mathbf{n} = \langle value \rangle$. These arguments must satisfy $\mathbf{tda} \geq \mathbf{n}$.

NE_INT_ARG_LT

On entry, $\mathbf{n} = \langle value \rangle$.
Constraint: $\mathbf{n} \geq 0$.

7 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 . denotes the spectral (two) norm.

8 Parallelism and Performance

Not applicable.

9 Further Comments

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

10 Example

To obtain the QR factorization of the 5 by 3 matrix

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

10.1 Program Text

```
/* nag_real_qr (f01qcc) Example Program.
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 * Mark 8 revised, 2004.
 */

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

#define A(I, J) a[(I) *tda + J]
int main(void)
{

  Integer  exit_status = 0, i, j, m, n, tda;
  NagError fail;
  double   *a = 0, *zeta = 0;

  INIT_FAIL(fail);

  printf("nag_real_qr (f01qcc) Example Program Results\n");
  scanf(" %*[^\n]"); /* skip headings in data file */
  scanf(" %*[^\n]");
  scanf("%ld%ld", &m, &n);
  if (n >= 0 && m >= n)
    {
      if (!(zeta = NAG_ALLOC(n, double)) ||
          !(a = NAG_ALLOC(m*n, double)))
        {
          printf("Allocation failure\n");
          exit_status = -1;
          goto END;
        }
      tda = n;
    }
  else
    {
      printf("Invalid n or m.\n");
      exit_status = 1;
      return exit_status;
    }
  scanf(" %*[^\n]"); /* skip next heading */
  for (i = 0; i < m; ++i)   /* Read matrix A */
    for (j = 0; j < n; ++j)
```

```
      scanf("%lf", &A(i, j));

  /* Find the QR factorization of A */
  /* nag_real_qr (f01qcc).
   * QR factorization of real m by n matrix (m >= n)
   */
  nag_real_qr(m, n, a, tda, zeta, &fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_real_qr (f01qcc).\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }

  printf("QR factorization of A\n\n");
  printf("Vector zeta\n");
  for (i = 0; i < n; ++i)
    printf(" %8.4f", zeta[i]);
  printf("\n\n");
  printf("Matrix A after factorization (upper triangular part is R)\n");
  for (i = 0; i < m; ++i)
    {
      for (j = 0; j < n; ++j)
        printf(" %8.4f", A(i, j));
      printf("\n");
    }
 END:
  NAG_FREE(zeta);
  NAG_FREE(a);
  return exit_status;
}
```

10.2 Program Data

```
nag_real_qr (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
```

10.3 Program Results

```
nag_real_qr (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
```