Guide and Reference

DGEQRF--General Matrix QR Factorization

This subroutine computes the ***QR*** factorization of a real general matrix

A = ***QR***

where:

> ***Q*** is an orthogonal matrix.
> For *m* >= *n*, ***R*** is an upper triangular matrix.
> For *m* < *n*, ***R*** is an upper trapezoidal matrix.

Table 119. Data Types

A, ***tau***, *work*	**Subroutine**
Long-precision real	DGEQRF

Syntax:

Fortran	CALL DGEQRF(*m*, *n*, *a*, *lda*, *tau*, *work*, *lwork*, *info*)
C and C++	dgeqrf (*m*, *n*, *a*, *lda*, *tau*, *work*, *lwork*, *info*);
PL/I	CALL DGEQRF (*m*, *n*, *a*, *lda*, *tau*, *work*, *lwork*, *info*);

On Entry:

m

> is the number of rows in matrix ***A*** used in the computation.
>
> Specified as: a fullword integer; *m* >= 0.

n

> is the number of columns in matrix ***A*** used in the computation.
>
> Specified as: a fullword integer; *n* >= 0.

a

> is the *m* by *n* general matrix ***A*** whose ***QR*** factorization is to be computed.

Specified as: an *lda* by (at least) *n* array, containing numbers of the data type indicated in Table 119.

lda
is the leading dimension of the array specified for *a*.

Specified as: a fullword integer; *lda* > 0 and *lda* >= *m*.

tau
See "On Return".

work
has the following meaning:

If *lwork* = 0, *work* is ignored.

If *lwork* <> 0, *work* is the work area used by this subroutine, where:

- If *lwork* <> -1, its size is (at least) of length *lwork*.
- If *lwork* = -1, its size is (at least) of length 1.

Specified as: an area of storage containing numbers of data type indicated in Table 119.

lwork
is the number of elements in array WORK.

Specified as: a fullword integer; where:

- If *lwork* = 0, DGEQRF dynamically allocates the work area used by this subroutine. The work area is deallocated before control is returned to the calling program. This option is an extension to the LAPACK standard.
- If *lwork* = -1, DGEQRF performs a work area query and returns the optimal size of *work* in $work_1$. No computation is performed and the subroutine returns after error checking is complete.
- Otherwise, it must be:

  lwork >= max(1, *n*)

info
See "On Return".

On Return:

a
is the updated general matrix ***A***, containing the results of the computation.

The elements on and above the diagonal of the array contain the min(*m*, *n*) × *n* upper trapezoidal matrix ***R*** (***R*** is upper triangular if *m* >= *n*). The elements below the diagonal with tau represent the

orthogonal matrix ***Q*** as a product of min(*m*, *n*) elementary reflectors.

Returned as: an *lda* by (at least) *n* array, containing numbers of the data type indicated in Table 119.

tau

is the vector ***tau***, of length min(*m*, *n*), containing the scalar factors of the elementary reflectors.

Returned as: a one-dimensional array of (at least) length min(*m*, *n*), containing numbers of the data type indicated in Table 119.

work

is the work area used by this subroutine if *lwork* <> 0, where:

If *lwork* <> 0 and *lwork* <> -1, its size is (at least) of length *lwork*.

If *lwork* = -1, its size is (at least) of length 1.

Returned as: an area of storage, where:

If *lwork* >= 1 or *lwork* = -1, then $work_1$ is set to the optimal *lwork* value and contains numbers of the data type indicated in Table 119. Except for $work_1$, the contents of *work* are overwritten on return.

info

indicates that a successful computation occurred.

Returned as: a fullword integer; *info* = 0.

Notes and Coding Rules:

1. In your C program, argument *info* must be passed by reference.
2. The vectors and matrices used in the computation must have no common elements; otherwise, results are unpredictable.
3. For best perfomance specify *lwork* = 0.

Function: Compute the ***QR*** factorization of a real general matrix ***A***

$\boldsymbol{A} = \boldsymbol{QR}$

where:

The orthogonal matrix ***Q*** is represented as a product of elementary reflectors:

$$\boldsymbol{Q} = \boldsymbol{H}_1 \boldsymbol{H}_2 \ldots \boldsymbol{H}_k$$

where:

$k = \min(m, n)$

For each i: $\boldsymbol{H}_i = \boldsymbol{I}$-tau$\boldsymbol{v}\boldsymbol{v}^T$

tau is a real scalar, stored on return in $\boldsymbol{tau}_i$

$\boldsymbol{v}$ is a real vector with $\boldsymbol{v}_{1:i-1} = 0$, $\boldsymbol{v}_i = 1$.

$\boldsymbol{v}_{i+1:m}$ is stored on return in $\boldsymbol{A}_{i+1:m,\, i}$

$\boldsymbol{I}$ is the identity matrix

For $m >= n$, $\boldsymbol{R}$ is an upper triangular matrix.
For $m < n$, $\boldsymbol{R}$ is an upper trapezoidal matrix.

If $m = 0$ or $n = 0$, no computation is performed and the subroutine returns after doing some parameter checking.

See references [52,8,65,50,51].

Error Conditions:

Resource Errors: *lwork* = 0 and unable to allocate work space.

Computational Errors: None.

Input-Argument Errors:

1. $m < 0$
2. $n < 0$
3. *lda <= 0*
4. *lda* < *m*
5. *lwork* <> 0, *lwork* <> -1, and *lwork* < max(1, *n*)

Example 1: This example shows the ***QR*** factorization of a general matrix ***A*** of size 6 × 2.
Note: Because *lwork* = 0, DGEQRF dynamically allocates the work area used by this subroutine.

Call Statements and Input:

```
              M    N    A  LDA  TAU   WORK  LWORK  INFO
              |    |    |   |    |     |      |      |
CALL DGEQRF ( 6 ,  2  , A , 6 , TAU , WORK ,  0  , INFO)
```

General matrix ***A*** of size 6 × 2:

```
        *                    *
        |   .000000  2.000000 |
        |  2.000000 -1.000000 |
A    =  |  2.000000 -1.000000 |
        |   .000000  1.500000 |
        |  2.000000 -1.000000 |
        |  2.000000 -1.000000 |
        *                    *
```

Output:

General matrix ***A*** of size 6 × 2.

```
        *                    *
        | -4.000000  2.000000 |
        |   .500000  2.500000 |
A   =   |   .500000   .285714 |
        |   .000000  -.428571 |
        |   .500000   .285714 |
        |   .500000   .285714 |
        *                    *
```

Vector ***tau*** of length 2:

```
        *                    *
TAU  =  |  1.000000  1.400000 |
        *                    *
```

The value of *info* is 0.

Example 2: This example shows the ***QR*** factorization of a general matrix ***A*** of size 4x5.
Note: Because *lwork* = 0, DGEQRF dynamically allocates the work area used by this subroutine.

Call Statements and Input:

```
              M    N    A  LDA  TAU   WORK  LWORK  INFO
              |    |    |   |    |     |      |      |
CALL DGEQRF ( 4 ,  5  , A , 4 , TAU , WORK ,  0  , INFO)
```

General matrix ***A*** of size 4×5:

```
        *                                              *
        |   .500000   .500000  1.207107   .000000  1.707107 |
A   =   |   .500000 -1.500000  -.500000  2.414214   .707107 |
        |   .500000   .500000   .207107   .000000   .292893 |
        |   .500000 -1.500000  -.500000  -.414214  -.707107 |
        *                                              *
```

Output:

General matrix ***A*** of size 4×5:

```
        *                                              *
        | -1.000000  1.000000  -.207107 -1.000000 -1.000000 |
A   =   |   .333333  2.000000  1.207107 -1.000000  1.000000 |
        |   .333333  -.200000   .707107   .000000  1.000000 |
        |   .333333   .400000   .071068 -2.000000 -1.000000 |
        *                                              *
```

Vector ***tau*** of length 4:

```
        *                                    *
TAU  =  |  1.500000  1.666667  1.989949   .000000 |
        *                                    *
```

The value of *info* is 0.

[Top of Page | Previous Page | Next Page | Table of Contents | Index]