NAG Library Routine Document

F06QPF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

F06QPF performs a QR factorization (as a sequence of plane rotations) of a real upper triangular matrix that has been modified by a rank-1 update.

2 Specification

```
SUBROUTINE F06QPF(N, ALPHA, X, INCX, Y, INCY, A, LDA, C, S)
INTEGER            N, INCX, INCY, LDA
double precision   ALPHA, X(*), Y(*), A(LDA,*), C(N-1), S(N-1)
```

3 Description

F06QPF performs a QR factorization of an upper triangular matrix which has been modified by a rank-1 update:

$$\alpha xy^{\mathrm{T}} + U = QR$$

where U and R are n by n real upper triangular matrices, x and y are n element real vectors, α is a real scalar, and Q is an n by n real orthogonal matrix.

Q is formed as the product of two sequences of plane rotations:

$$Q^{\mathrm{T}} = Q_{n-1} \cdots Q_2 Q_1 P_1 P_2 \cdots P_{n-1}$$

where

P_k is a rotation in the (k,n) plane, chosen to annihilate x_k: thus $Px = \beta e_n$, where $P = P_1 P_2 \cdots P_{n-1}$ and e_n is the last column of the unit matrix;

Q_k is a rotation in the (k,n) plane, chosen to annihilate the (n,k) element of $\left(\alpha\beta e_n y^{\mathrm{T}} + PU\right)$, and thus restore it to upper triangular form.

The 2 by 2 plane rotation part of P_k or Q_k has the form

$$\begin{pmatrix} c_k & s_k \\ -s_k & c_k \end{pmatrix}.$$

The tangents of the rotations P_k are returned in the array X; the cosines and sines of these rotations can be recovered by calling F06BCF. The cosines and sines of the rotations Q_k are returned directly in the arrays C and S.

4 References

None.

5 Parameters

1: N – INTEGER *Input*

On entry: n, the order of the matrices U and R.

Constraint: $\mathrm{N} \geq 0$.

2: ALPHA – ***double precision*** *Input*

On entry: the scalar α.

3: X($*$) – ***double precision*** array *Input/Output*

Note: the dimension of the array X must be at least $1 + (\mathrm{N} - 1)|\mathrm{INCX}|$.

On entry: the incremented array X must contain the n element vector x.

On exit: the tangents of the rotations P_k, for $k = 1, 2, \ldots, n - 1$.

4: INCX – INTEGER *Input*

On entry: the increment in the subscripts of X between successive elements of x.

Constraint: INCX > 0.

5: Y($*$) – ***double precision*** array *Input*

Note: the dimension of the array Y must be at least $1 + (\mathrm{N} - 1)|\mathrm{INCY}|$.

On entry: the incremented array Y must contain the n element vector y.

6: INCY – INTEGER *Input*

On entry: the increment in the subscripts of Y between successive elements of y.

Constraint: INCY > 0.

7: A(LDA,$*$) – ***double precision*** array *Input/Output*

Note: the second dimension of the array A must be at least N.

On entry: the n by n upper triangular matrix U.

On exit: the upper triangular matrix R.

8: LDA – INTEGER *Input*

On entry: the first dimension of the array A as declared in the (sub)program from which F06QPF is called.

Constraint: LDA $\geq \max(1, \mathrm{N})$.

9: C(N – 1) – ***double precision*** array *Output*

On exit: the cosines of the rotations Q_k, for $k = 1, 2, \ldots, n - 1$.

10: S(N – 1) – ***double precision*** array *Output*

On exit: the sines of the rotations Q_k, for $k = 1, 2, \ldots, n - 1$.

6 Error Indicators and Warnings

None.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

None.