

NAG Library Routine Document

C06FRF

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

C06FRF computes the discrete Fourier transforms of m sequences, each containing n complex data values. This routine is designed to be particularly efficient on vector processors.

2 Specification

```
SUBROUTINE C06FRF(M, N, X, Y, INIT, TRIG, WORK, IFAIL)
INTEGER          M, N, IFAIL
double precision X(M*N), Y(M*N), TRIG(2*N), WORK(2*M*N)
CHARACTER*1      INIT
```

3 Description

Given m sequences of n complex data values z_j^p , for $j = 0, 1, \dots, n-1$ and $p = 1, 2, \dots, m$, C06FRF simultaneously calculates the Fourier transforms of all the sequences defined by

$$\hat{z}_k^p = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} z_j^p \times \exp\left(-i \frac{2\pi jk}{n}\right), \quad k = 0, 1, \dots, n-1 \text{ and } p = 1, 2, \dots, m.$$

(Note the scale factor $\frac{1}{\sqrt{n}}$ in this definition.)

The discrete Fourier transform is sometimes defined using a positive sign in the exponential term

$$\hat{z}_k^p = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} z_j^p \times \exp\left(+i \frac{2\pi jk}{n}\right).$$

To compute this form, this routine should be preceded and followed by a call of C06GCF to form the complex conjugates of the z_j^p and the $\hat{z}_k^p$.

The routine uses a variant of the fast Fourier transform (FFT) algorithm (see Brigham (1974)) known as the Stockham self-sorting algorithm, which is described in Temperton (1983). Special code is provided for the factors 2, 3, 4, 5 and 6. This routine is designed to be particularly efficient on vector processors, and it becomes especially fast as m , the number of transforms to be computed in parallel, increases.

4 References

Brigham E O (1974) *The Fast Fourier Transform* Prentice-Hall

Temperton C (1983) Self-sorting mixed-radix fast Fourier transforms *J. Comput. Phys.* **52** 1–23

5 Parameters

1: M – INTEGER

Input

On entry: m , the number of sequences to be transformed.

Constraint: $M \geq 1$.

- 2: N – INTEGER Input
On entry: n , the number of complex values in each sequence.
Constraint: $N \geq 1$.
- 3: X($M \times N$) – **double precision** array Input/Output
 4: Y($M \times N$) – **double precision** array Input/Output
On entry: the real and imaginary parts of the complex data must be stored in X and Y respectively as if in a two-dimensional array of dimension (1 : M, 0 : N – 1); each of the m sequences is stored in a **row** of each array. In other words, if the real parts of the p th sequence to be transformed are denoted by x_j^p , for $j = 0, 1, \dots, n - 1$, then the mn elements of the array X must contain the values
- $$x_0^1, x_0^2, \dots, x_0^m, x_1^1, x_1^2, \dots, x_1^m, \dots, x_{n-1}^1, x_{n-1}^2, \dots, x_{n-1}^m.$$
- On exit:* X and Y are overwritten by the real and imaginary parts of the complex transforms.
- 5: INIT – CHARACTER*1 Input
On entry: indicates whether trigonometric coefficients are to be calculated.
 INIT = 'I'
 Calculate the required trigonometric coefficients for the given value of n , and store in the array TRIG.
 INIT = 'S' or 'R'
 The required trigonometric coefficients are assumed to have been calculated and stored in the array TRIG in a prior call to one of C06FPF, C06FQF or C06FRF. The routine performs a simple check that the current value of n is consistent with the values stored in TRIG.
Constraint: INIT = 'I', 'S' or 'R'.
- 6: TRIG($2 \times N$) – **double precision** array Input/Output
On entry: if INIT = 'S' or 'R', TRIG must contain the required trigonometric coefficients that have been previously calculated. Otherwise TRIG need not be set.
On exit: contains the required coefficients (computed by the routine if INIT = 'I').
- 7: WORK($2 \times M \times N$) – **double precision** array Workspace
- 8: IFAIL – INTEGER Input/Output
On entry: IFAIL must be set to 0, –1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.
On exit: IFAIL = 0 unless the routine detects an error (see Section 6).
 For environments where it might be inappropriate to halt program execution when an error is detected, the value –1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value –1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or –1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, $M < 1$.

IFAIL = 2

On entry, $N < 1$.

IFAIL = 3

On entry, INIT $\neq$ 'I', 'S' or 'R'.

IFAIL = 4

Not used at this Mark.

IFAIL = 5

On entry, INIT = 'S' or 'R', but the array TRIG and the current value of N are inconsistent.

IFAIL = 6

An unexpected error has occurred in an internal call. Check all subroutine calls and array dimensions. Seek expert help.

7 Accuracy

Some indication of accuracy can be obtained by performing a subsequent inverse transform and comparing the results with the original sequence (in exact arithmetic they would be identical).

8 Further Comments

The time taken by C06FRF is approximately proportional to $nm \log n$, but also depends on the factors of n . C06FRF is fastest if the only prime factors of n are 2, 3 and 5, and is particularly slow if n is a large prime, or has large prime factors.

9 Example

This example reads in sequences of complex data values and prints their discrete Fourier transforms (as computed by C06FRF). Inverse transforms are then calculated using C06FRF and C06GCF and printed out, showing that the original sequences are restored.

9.1 Program Text

```
*      C06FRF Example Program Text
*      Mark 14 Revised. NAG Copyright 1989.
*      .. Parameters ..
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=5,NMAX=20)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*      .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
*      .. Local Arrays ..
      DOUBLE PRECISION TRIG(2*NMAX), WORK(2*MMAX*NMAX), X(MMAX*NMAX),
+      Y(MMAX*NMAX)
*      .. External Subroutines ..
      EXTERNAL         C06FRF, C06GCF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'C06FRF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
20  READ (NIN,*,END=120) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
        DO 40 J = 1, M
          READ (NIN,*) (X(I*M+J),I=0,N-1)
          READ (NIN,*) (Y(I*M+J),I=0,N-1)
40    CONTINUE
        WRITE (NOUT,*)
```

```

      WRITE (NOUT,*) 'Original data values'
      DO 60 J = 1, M
        WRITE (NOUT,*)
        WRITE (NOUT,99999) 'Real ', (X(I*M+J),I=0,N-1)
        WRITE (NOUT,99999) 'Imag ', (Y(I*M+J),I=0,N-1)
60    CONTINUE
      IFAIL = 1
*
      CALL C06FRF(M,N,X,Y,'Initial',TRIG,WORK,IFAIL)
*
      IF (IFAIL.EQ.0) THEN
        WRITE (NOUT,*)
        WRITE (NOUT,*) 'Discrete Fourier transforms'
        DO 80 J = 1, M
          WRITE (NOUT,*)
          WRITE (NOUT,99999) 'Real ', (X(I*M+J),I=0,N-1)
          WRITE (NOUT,99999) 'Imag ', (Y(I*M+J),I=0,N-1)
80    CONTINUE
*
        CALL C06GCF(Y,M*N,IFAIL)
        CALL C06FRF(M,N,X,Y,'Subsequent',TRIG,WORK,IFAIL)
        CALL C06GCF(Y,M*N,IFAIL)
*
        WRITE (NOUT,*)
        WRITE (NOUT,*)
        +      'Original data as restored by inverse transform'
        DO 100 J = 1, M
          WRITE (NOUT,*)
          WRITE (NOUT,99999) 'Real ', (X(I*M+J),I=0,N-1)
          WRITE (NOUT,99999) 'Imag ', (Y(I*M+J),I=0,N-1)
100    CONTINUE
        GO TO 20
      ELSE
        WRITE (NOUT,*)
        WRITE (NOUT,99998) ' ** C06FRF returned with IFAIL = ',
        +      IFAIL
        END IF
      ELSE
        WRITE (NOUT,*) 'Invalid value of M or N'
      END IF
120 CONTINUE
*
99999 FORMAT (1X,A,6F10.4)
99998 FORMAT (1X,A,I5)
      END

```

9.2 Program Data

C06FRF Example Program Data

3	6					
0.3854	0.6772	0.1138	0.6751	0.6362	0.1424	
0.5417	0.2983	0.1181	0.7255	0.8638	0.8723	
0.9172	0.0644	0.6037	0.6430	0.0428	0.4815	
0.9089	0.3118	0.3465	0.6198	0.2668	0.1614	
0.1156	0.0685	0.2060	0.8630	0.6967	0.2792	
0.6214	0.8681	0.7060	0.8652	0.9190	0.3355	

9.3 Program Results

C06FRF Example Program Results

Original data values

Real	0.3854	0.6772	0.1138	0.6751	0.6362	0.1424
Imag	0.5417	0.2983	0.1181	0.7255	0.8638	0.8723
Real	0.9172	0.0644	0.6037	0.6430	0.0428	0.4815
Imag	0.9089	0.3118	0.3465	0.6198	0.2668	0.1614
Real	0.1156	0.0685	0.2060	0.8630	0.6967	0.2792

Imag	0.6214	0.8681	0.7060	0.8652	0.9190	0.3355
------	--------	--------	--------	--------	--------	--------

Discrete Fourier transforms

Real	1.0737	-0.5706	0.1733	-0.1467	0.0518	0.3625
Imag	1.3961	-0.0409	-0.2958	-0.1521	0.4517	-0.0321

Real	1.1237	0.1728	0.4185	0.1530	0.3686	0.0101
Imag	1.0677	0.0386	0.7481	0.1752	0.0565	0.1403

Real	0.9100	-0.3054	0.4079	-0.0785	-0.1193	-0.5314
Imag	1.7617	0.0624	-0.0695	0.0725	0.1285	-0.4335

Original data as restored by inverse transform

Real	0.3854	0.6772	0.1138	0.6751	0.6362	0.1424
Imag	0.5417	0.2983	0.1181	0.7255	0.8638	0.8723

Real	0.9172	0.0644	0.6037	0.6430	0.0428	0.4815
Imag	0.9089	0.3118	0.3465	0.6198	0.2668	0.1614

Real	0.1156	0.0685	0.2060	0.8630	0.6967	0.2792
Imag	0.6214	0.8681	0.7060	0.8652	0.9190	0.3355
