C06FAF – NAG Fortran Library Routine Document

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

C06FAF calculates the discrete Fourier transform of a sequence of n real data values (using a work array for extra speed).

2 Specification

```
SUBROUTINE C06FAF(X, N, WORK, IFAIL)
INTEGER           N, IFAIL
real              X(N), WORK(N)
```

3 Description

Given a sequence of n real data values x_j, for $j = 0,1,\ldots,n-1$, this routine calculates their discrete Fourier transform defined by:

$$\hat{z}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \times \exp\left(-i\frac{2\pi jk}{n}\right), \qquad k = 0,1,\ldots,n-1.$$

(Note the scale factor of $\frac{1}{\sqrt{n}}$ in this definition.) The transformed values $\hat{z}_k$ are complex, but they form a Hermitian sequence (i.e., $\hat{z}_{n-k}$ is the complex conjugate of $\hat{z}_k$), so they are completely determined by n real numbers (see also the Chapter Introduction).

To compute the inverse discrete Fourier transform defined by:

$$\hat{w}_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} x_j \times \exp\left(+i\frac{2\pi jk}{n}\right),$$

this routine should be followed by a call of C06GBF to form the complex conjugates of the $\hat{z}_k$.

The routine uses the fast Fourier transform (FFT) algorithm in Brigham [1]. There are some restrictions on the value of n (see Section 5).

4 References

[1] Brigham E O (1973) *The Fast Fourier Transform* Prentice–Hall

5 Parameters

1: X(N) — ***real*** array *Input/Output*

On entry: if X is declared with bounds (0:N−1) in the (sub)program from which C06FAF is called, then X(j) must contain x_j, for $j = 0,1,\ldots,n-1$.

On exit: the discrete Fourier transform stored in Hermitian form. If the components of the transform $\hat{z}_k$ are written as $a_k + ib_k$, and if X is declared with bounds (0:N−1) in the (sub)program from which C06FAF is called, then for $0 \le k \le n/2$, a_k is contained in X(k), and for $1 \le k \le (n-1)/2$, b_k is contained in X($n-k$). (See also Section 2.1.2 of the Chapter Introduction and the Example Program.)

2: N — INTEGER *Input*

On entry: the number of data values, n. The largest prime factor of N must not exceed 19, and the total number of prime factors of N, counting repetitions, must not exceed 20.

Constraint: N > 1.

3: WORK(N) — ***real*** array *Workspace*

4: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

At least one of the prime factors of N is greater than 19.

IFAIL = 2

N has more than 20 prime factors.

IFAIL = 3

$N \leq 1$.

IFAIL = 4

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 the routine is approximately proportional to $n \times \log n$, but also depends on the factorization of n. The routine is somewhat faster than average if the only prime factors of n are 2, 3 or 5; and fastest of all if n is a power of 2.

9 Example

This program reads in a sequence of real data values and prints their discrete Fourier transform (as computed by C06FAF), after expanding it from Hermitian form into a full complex sequence.

It then performs an inverse transform, using C06GBF and C06FBF, and prints the sequence so obtained alongside the original data values.

9.1 Program Text

Note. The listing of the example program presented below uses bold italicised terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     C06FAF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NMAX
      PARAMETER        (NMAX=20)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
```

```
      INTEGER          IFAIL, J, N, N2, NJ
*     .. Local Arrays ..
      real             A(0:NMAX-1), B(0:NMAX-1), WORK(NMAX),
     +                 X(0:NMAX-1), XX(0:NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         C06FAF, C06FBF, C06GBF
*     .. Intrinsic Functions ..
      INTRINSIC        MOD
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06FAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 READ (NIN,*,END=120) N
      IF (N.GT.1 .AND. N.LE.NMAX) THEN
         DO 40 J = 0, N - 1
            READ (NIN,*) X(J)
            XX(J) = X(J)
   40    CONTINUE
         IFAIL = 0
*
         CALL C06FAF(X,N,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Components of discrete Fourier transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '          Real      Imag'
         WRITE (NOUT,*)
         A(0) = X(0)
         B(0) = 0.0e0
         N2 = (N-1)/2
         DO 60 J = 1, N2
            NJ = N - J
            A(J) = X(J)
            A(NJ) = X(J)
            B(J) = X(NJ)
            B(NJ) = -X(NJ)
   60    CONTINUE
         IF (MOD(N,2).EQ.0) THEN
            A(N2+1) = X(N2+1)
            B(N2+1) = 0.0e0
         END IF
         DO 80 J = 0, N - 1
            WRITE (NOUT,99999) J, A(J), B(J)
   80    CONTINUE
*
         CALL C06GBF(X,N,IFAIL)
         CALL C06FBF(X,N,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence as restored by inverse transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '        Original  Restored'
         WRITE (NOUT,*)
         DO 100 J = 0, N - 1
            WRITE (NOUT,99999) J, XX(J), X(J)
  100    CONTINUE
         GO TO 20
```

```
      ELSE
         WRITE (NOUT,*) 'Invalid value of N'
      END IF
  120 STOP
*
99999 FORMAT (1X,I5,2F10.5)
      END
```

9.2 Program Data

```
C06FAF Example Program Data
    7
  0.34907
  0.54890
  0.74776
  0.94459
  1.13850
  1.32850
  1.51370
```

9.3 Program Results

```
 C06FAF Example Program Results

 Components of discrete Fourier transform

           Real      Imag

     0   2.48361   0.00000
     1  -0.26599   0.53090
     2  -0.25768   0.20298
     3  -0.25636   0.05806
     4  -0.25636  -0.05806
     5  -0.25768  -0.20298
     6  -0.26599  -0.53090

 Original sequence as restored by inverse transform

         Original  Restored

     0   0.34907   0.34907
     1   0.54890   0.54890
     2   0.74776   0.74776
     3   0.94459   0.94459
     4   1.13850   1.13850
     5   1.32850   1.32850
     6   1.51370   1.51370
```