# NAG Fortran Library Routine Document

# C06PAF

**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

C06PAF calculates the discrete Fourier transform of a sequence of $n$ real data values or of a Hermitian sequence of $n$ complex data values.

## 2 Specification

```
SUBROUTINE C06PAF(DIRECT, X, N, WORK, IFAIL)
INTEGER           N, IFAIL
real              X(N+2), WORK(3*N+15)
CHARACTER*1       DIRECT
```

## 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 (in the **Forward** direction) defined by

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

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 (since $\hat{z}_0$ is real, as is $\hat{z}_{n/2}$ for $n$ even).

Alternatively, given a Hermitian sequence of $n$ complex data values $z_j$, this routine calculates their inverse (**backward**) discrete Fourier transform defined by

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

The transformed values $\hat{x}_k$ are real.

(Note the scale factor of $\frac{1}{\sqrt{n}}$ in the above definitions.) A call of the routine with DIRECT = 'F' followed by a call with DIRECT = 'B' will restore the original data.

The routine uses a variant of the fast Fourier transform (FFT) algorithm (Brigham (1974)) known as the Stockham self-sorting algorithm, which is described in Temperton (1983b).

## 4 References

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

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

# 5 Parameters

1: DIRECT – CHARACTER*1 *Input*

*On entry*: if the **F**orward transform as defined in Section 3 is to be computed, then DIRECT must be set equal to 'F'. If the **B**ackward transform is to be computed then DIRECT must be set equal to 'B'.

*Constraint*: DIRECT = 'F' or 'B'.

2: X(N+2) – ***real*** array *Input/Output*

*On entry*: if X is declared with bounds $(0 : \mathrm{N} + 1)$ in the (sub)program from which C06PAF is called, then:

if DIRECT is set to 'F', X$(j)$ must contain $x_j$, for $j = 0, 1, \ldots, n-1$;

if DIRECT is set to 'B', X$(2 * k)$ and X$(2 * k + 1)$ must contain the real and imaginary parts respectively of $\hat{z}_k$, for $k = 0, 1, \ldots, n/2$. (Note that for the sequence $\hat{z}_k$ to be Hermitian, the imaginary part of $\hat{z}_0$, and of $\hat{z}_{n/2}$ for $n$ even, must be zero.)

*On exit*:

if DIRECT is set to 'F' and X is declared with bounds $(0 : \mathrm{N} + 1)$ then X$(2 * k)$ and X$(2 * k + 1)$ will contain the real and imaginary parts respectively of $\hat{z}_k$, for $k = 0, 1, \ldots, n/2$;

if DIRECT is set to 'B' and X is declared with bounds $(0 : \mathrm{N} + 1)$ then X$(j)$ will contain $x_j$, for $j = 0, 1, \ldots, n-1$.

3: N – INTEGER *Input*

*On entry*: the number of data values, $n$. The total number of prime factors of N, counting repetitions, must not exceed 30.

*Constraint*: N $> 1$.

4: WORK(3∗N+15) – ***real*** array *Workspace*

5: IFAIL – INTEGER *Input/Output*

*On entry*: IFAIL must be set to $0$, $-1$ or $1$. Users who are unfamiliar with this parameter should refer to Chapter P01 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, for users 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, N $\leq 1$.

IFAIL = 2

IFAIL = 3

On entry, at least one of the prime factors of N is greater than 19.

IFAIL = 4

On entry, N has more than 30 prime factors.

IFAIL = 5

## 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 C06PAF with DIRECT set to 'F'), after expanding it from complex Hermitian form into a full complex sequence.

It then performs an inverse transform, using C06PAF with DIRECT set to 'B', and prints the sequence 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.

```
*     C06PAF Example Program Text.
*     Mark 19 Release. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX
      PARAMETER        (NMAX=20)
*     .. Local Scalars ..
      INTEGER          IFAIL, J, N, NJ
*     .. Local Arrays ..
      real             WORK(2*NMAX+15), X(0:NMAX+1), XX(0:NMAX-1)
*     .. External Subroutines ..
      EXTERNAL         C06PAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06PAF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 CONTINUE
      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 C06PAF('F',X,N,WORK,IFAIL)
```

```
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Components of discrete Fourier transform'
         WRITE (NOUT,*)
         WRITE (NOUT,*) '          Real      Imag'
         WRITE (NOUT,*)
         DO 60 J = 0, N/2
            WRITE (NOUT,99999) J, X(2*J), X(2*J+1)
   60    CONTINUE
         DO 80 J = N/2 + 1, N - 1
            NJ = N - J
            WRITE (NOUT,99999) J, X(2*NJ), -X(2*NJ+1)
   80    CONTINUE
*
         CALL C06PAF('B',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 CONTINUE
      STOP
*
99999 FORMAT (1X,I5,2F10.5)
      END
```

## 9.2 Program Data

```
C06PAF Example Program Data
    7
  0.34907
  0.54890
  0.74776
  0.94459
  1.13850
  1.32850
  1.51370
```

## 9.3 Program Results

```
 C06PAF 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
```