C06PPF – 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

C06PPF computes the discrete Fourier transforms of m sequences, each containing n real data values or a Hermitian complex sequence stored in a complex storage format.

2 Specification

```
SUBROUTINE C06PPF(DIRECT, M, N, X, WORK, IFAIL)
CHARACTER*1       DIRECT
INTEGER           M, N, IFAIL
real              X(M*(N+2)), WORK(M*N+2*N+2*M+15)
```

3 Description

Given m sequences of n real data values x_j^p, for $j = 0, 1, \ldots, n-1$ and $p = 1, 2, \ldots, m$, this routine simultaneously calculates the Fourier transforms of all the sequences defined by

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

The transformed values $\hat{z}_k^p$ are complex, but for each value of p the $\hat{z}_k^p$ form a Hermitian sequence (i.e., $\hat{z}_{n-k}^p$ is the complex conjugate of $\hat{z}_k^p$), so they are completely determined by mn real numbers (since $\hat{z}_0^p$ is real, as is $\hat{z}_{n/2}^p$ for n even).

Alternatively, given m Hermitian sequences of n complex data values z_j^p, this routine simultaneously calculates their inverse (**backward**) discrete Fourier transforms defined by

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

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

(Note the scale factor $\frac{1}{\sqrt{n}}$ in the above definition.) 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 [1]) known as the Stockham self-sorting algorithm, which is described in Temperton [2]. Special coding is provided for the factors 2, 3, 4 and 5.

4 References

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

[2] Temperton C (1983) Fast mixed-radix real Fourier transforms *J. Comput. Phys.* **52** 340–350

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: M — INTEGER *Input*

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

Constraint: $M \geq 1$.

3: N — INTEGER *Input*

On entry: the number of real or complex values in each sequence, n.

Constraint: $N \geq 1$.

4: X(M∗(N+2)) — ***real*** array *Input/Output*

On entry: the data must be stored in X 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 the array. In other words, if the data values of the pth sequence to be transformed are denoted by x_j^p, for $j = 0, 1, \ldots, n-1$, then:

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

On exit:

if DIRECT is set to 'F' and X is declared with bounds (1:M,0:N+1) then X(p,2∗k) and X(p,2∗k+1) will contain the real and imaginary parts respectively of $\hat{z}_k^p$, for $k = 0, 1, \ldots, n/2$ and $p = 1, 2, \ldots, m$;
if DIRECT is set to 'B' and X is declared with bounds (1:M,0:N+1) then X(p, j) will contain x_j^p, for $j = 0, 1, \ldots, n-1$ and $p = 1, 2, \ldots, m$.

5: WORK(M∗N+2∗N+2∗M+15) — ***real*** array *Workspace*

The workspace requirements as documented for this routine may be an overestimate in some implementations. For full details of the workspace required by this routine please refer to the Users' Note for your implementation.

On exit: WORK(1) contains the minimum workspace required for the current values of M and N with this implementation.

6: 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

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

Errors detected by the routine:

IFAIL = 1

On entry, $M < 1$.

IFAIL = 2

On entry, $N < 1$.

IFAIL = 3

On entry, DIRECT not equal to one of 'F' or 'B'.

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 $nm \times \log n$, but also depends on the factors of n. The routine 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 program reads in sequences of real data values and prints their discrete Fourier transforms (as computed by C06PPF with DIRECT set to 'F'), after expanding them from complex Hermitian form into a full complex sequences.

Inverse transforms are then calculated by calling C06PPF with DIRECT set to 'B' showing that the original sequences are restored.

9.1 Program Text

```
*     C06PPF Example Program Text.
*     Mark 19 Release. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX
      PARAMETER        (MMAX=5,NMAX=20)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, J, M, N
*     .. Local Arrays ..
      real             WORK((MMAX+2)*(NMAX+2)+11), X((NMAX+2)*MMAX)
*     .. External Subroutines ..
      EXTERNAL         C06PPF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06PPF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 CONTINUE
      READ (NIN,*,END=140) 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)
   40    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Original data values'
         WRITE (NOUT,*)
         DO 60 J = 1, M
```

```
            WRITE (NOUT,99999) '     ', (X(I*M+J),I=0,N-1)
   60    CONTINUE
         IFAIL = 0
*
         CALL C06PPF('F',M,N,X,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Discrete Fourier transforms in complex Hermitian format'
         DO 80 J = 1, M
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Real ', (X(2*I*M+J),I=0,N/2)
            WRITE (NOUT,99999) 'Imag ', (X((2*I+1)*M+J),I=0,N/2)
   80    CONTINUE
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Fourier transforms in full complex form'
*
*
         DO 100 J = 1, M
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'Real ', (X(2*I*M+J),I=0,N/2),
     +        (X(2*(N-I)*M+J),I=N/2+1,N-1)
            WRITE (NOUT,99999) 'Imag ', (X((2*I+1)*M+J),I=0,N/2),
     +        (-X((2*(N-I)+1)*M+J),I=N/2+1,N-1)
  100    CONTINUE
*
         CALL C06PPF('B',M,N,X,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Original data as restored by inverse transform'
         WRITE (NOUT,*)
         DO 120 J = 1, M
            WRITE (NOUT,99999) '     ', (X(I*M+J),I=0,N-1)
  120    CONTINUE
         GO TO 20
      ELSE
         WRITE (NOUT,*) 'Invalid value of M or N'
      END IF
  140 CONTINUE
      STOP
*
99999 FORMAT (1X,A,9(:1X,F10.4))
      END
```

9.2 Program Data

```
C06PPF 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
```

9.3 Program Results

```
C06PPF Example Program Results

Original data values

          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

Discrete Fourier transforms in complex Hermitian format

Real      1.0737    -0.1041     0.1126    -0.1467
Imag      0.0000    -0.0044    -0.3738     0.0000

Real      1.3961    -0.0365     0.0780    -0.1521
Imag      0.0000     0.4666    -0.0607     0.0000

Real      1.1237     0.0914     0.3936     0.1530
Imag      0.0000    -0.0508     0.3458     0.0000

Fourier transforms in full complex form

Real      1.0737    -0.1041     0.1126    -0.1467     0.1126    -0.1041
Imag      0.0000    -0.0044    -0.3738     0.0000     0.3738     0.0044

Real      1.3961    -0.0365     0.0780    -0.1521     0.0780    -0.0365
Imag      0.0000     0.4666    -0.0607     0.0000     0.0607    -0.4666

Real      1.1237     0.0914     0.3936     0.1530     0.3936     0.0914
Imag      0.0000    -0.0508     0.3458     0.0000    -0.3458     0.0508

Original data as restored by inverse transform

          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
```