NAG Fortran Library Routine Document

C06PUF

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

C06PUF computes the two-dimensional discrete Fourier transform of a bivariate sequence of complex data values (using complex data type).

2 Specification

```
SUBROUTINE C06PUF(DIRECT, M, N, X, WORK, IFAIL)
INTEGER            M, N, IFAIL
complex            X(M*N), WORK(M*N+N+M+30)
CHARACTER*1        DIRECT
```

3 Description

This routine computes the two-dimensional discrete Fourier transform of a bivariate sequence of complex data values $z_{j_1j_2}$, where $j_1 = 0, 1, \ldots, m-1$ and $j_2 = 0, 1, \ldots, n-1$.

The discrete Fourier transform is here defined by

$$\hat{z}_{k_1k_2} = \frac{1}{\sqrt{mn}} \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{n-1} z_{j_1j_2} \times \exp\left(\pm 2\pi i \left(\frac{j_1k_1}{m} + \frac{j_2k_2}{n}\right)\right),$$

where $k_1 = 0, 1, \ldots, m-1$ and $k_2 = 0, 1, \ldots, n-1$.

(Note the scale factor of $\frac{1}{\sqrt{mn}}$ in this definition.) The minus sign is taken in the argument of the exponential within the summation when the forward transform is required, and the plus sign is taken when the backward transform is required. A call of the routine with DIRECT = 'F' followed by a call with DIRECT = 'B' will restore the original data.

This routine calls C06PRF to perform multiple one-dimensional discrete Fourier transforms by the fast Fourier transform (FFT) algorithm in Brigham (1974).

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: M – INTEGER *Input*

On entry: the first dimension of the transform, m.

Constraint: M $\geq$ 1.

3: N – INTEGER *Input*

On entry: the second dimension of the transform, n.

Constraint: N $\geq$ 1.

4: X(M*N) – ***complex*** array *Input/Output*

On entry: the complex data values. If X is regarded as a two-dimensional array of dimension $(0:\mathrm{M}-1,\ 0:\mathrm{N}-1)$, then $\mathrm{X}(j_1, j_2)$ must contain $z_{j_1 j_2}$.

On exit: the corresponding elements of the computed transform.

5: WORK(M*N+N+M+30) – ***complex*** array *Workspace*

6: 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

IFAIL = 2

IFAIL = 3

IFAIL = 4

IFAIL = 5

On entry, M has more than 30 prime factors.

IFAIL = 6

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 $mn \times \log(mn)$, but also depends on the factorization of the individual dimensions m and n. The routine is somewhat faster than average if their only prime factors are 2, 3 or 5; and fastest of all if they are powers of 2.

9 Example

This program reads in a bivariate sequence of complex data values and prints the two-dimensional Fourier transform. It then performs an inverse transform and prints the sequence so obtained, which may be compared to 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.

```
*     C06PUF Example Program Text.
*     Mark 19 Release. NAG Copyright 1999.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, MNMAX
      PARAMETER        (MMAX=96,NMAX=96,MNMAX=MMAX*NMAX)
*     .. Local Scalars ..
      INTEGER          IFAIL, M, N
*     .. Local Arrays ..
      complex          WORK(MMAX+NMAX+MNMAX+30), X(MNMAX)
*     .. External Subroutines ..
      EXTERNAL         C06PUF, READX, WRITX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C06PUF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
   20 CONTINUE
      READ (NIN,*,END=40) M, N
      IF (M*N.GE.1 .AND. M*N.LE.MNMAX) THEN
         CALL READX(NIN,X,M,N)
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Original data values'
         CALL WRITX(NOUT,X,M,N)
         IFAIL = 0
*
*        Compute transform
         CALL C06PUF('F',M,N,X,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*) 'Components of discrete Fourier transform'
         CALL WRITX(NOUT,X,M,N)
*
*        Compute inverse transform
         CALL C06PUF('B',M,N,X,WORK,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,*)
     +     'Original sequence as restored by inverse transform'
         CALL WRITX(NOUT,X,M,N)
         GO TO 20
      ELSE
         WRITE (NOUT,*) ' ** Invalid value of M or N'
      END IF
   40 CONTINUE
      STOP
      END
*
      SUBROUTINE READX(NIN,X,N1,N2)
*     Read 2-dimensional complex data
*     .. Scalar Arguments ..
      INTEGER          N1, N2, NIN
*     .. Array Arguments ..
      complex          X(N1,N2)
*     .. Local Scalars ..
      INTEGER          I, J
*     .. Executable Statements ..
      DO 20 I = 1, N1
```

```
         READ (NIN,*) (X(I,J),J=1,N2)
   20 CONTINUE
      RETURN
      END
*
      SUBROUTINE WRITX(NOUT,X,N1,N2)
*     Print 2-dimensional complex data
*     .. Scalar Arguments ..
      INTEGER          N1, N2, NOUT
*     .. Array Arguments ..
      complex          X(N1,N2)
*     .. Local Scalars ..
      INTEGER          I, J
*     .. Intrinsic Functions ..
      INTRINSIC        real, imag
*     .. Executable Statements ..
      DO 20 I = 1, N1
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Real ', (real(X(I,J)),J=1,N2)
         WRITE (NOUT,99999) 'Imag ', (imag(X(I,J)),J=1,N2)
   20 CONTINUE
      RETURN
*
99999 FORMAT (1X,A,7F10.3,/(6X,7F10.3))
      END
```

9.2 Program Data

```
C06PUF Example Program Data
3  5  : Number of rows, M, and columns, N, in X and Y
    ( 1.000, 0.000)
    ( 0.999,-0.040)
    ( 0.987,-0.159)
    ( 0.936,-0.352)
    ( 0.802,-0.597)
    ( 0.994,-0.111)
    ( 0.989,-0.151)
    ( 0.963,-0.268)
    ( 0.891,-0.454)
    ( 0.731,-0.682)
    ( 0.903,-0.430)
    ( 0.885,-0.466)
    ( 0.823,-0.568)
    ( 0.694,-0.720)
    ( 0.467,-0.884)
```

9.3 Program Results

```
 C06PUF Example Program Results

 Original data values

 Real      1.000     0.999     0.987     0.936     0.802
 Imag      0.000    -0.040    -0.159    -0.352    -0.597

 Real      0.994     0.989     0.963     0.891     0.731
 Imag     -0.111    -0.151    -0.268    -0.454    -0.682

 Real      0.903     0.885     0.823     0.694     0.467
 Imag     -0.430    -0.466    -0.568    -0.720    -0.884

 Components of discrete Fourier transform

 Real      3.373     0.481     0.251     0.054    -0.419
 Imag     -1.519    -0.091     0.178     0.319     0.415

 Real      0.457     0.055     0.009    -0.022    -0.076
 Imag      0.137     0.032     0.039     0.036     0.004

 Real     -0.170    -0.037    -0.042    -0.038    -0.002
```

```
Imag      0.493     0.058     0.008    -0.025    -0.083

Original sequence as restored by inverse transform

Real      1.000     0.999     0.987     0.936     0.802
Imag      0.000    -0.040    -0.159    -0.352    -0.597

Real      0.994     0.989     0.963     0.891     0.731
Imag     -0.111    -0.151    -0.268    -0.454    -0.682

Real      0.903     0.885     0.823     0.694     0.467
Imag     -0.430    -0.466    -0.568    -0.720    -0.884
```