NAG 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 (KIND=nag_wp)  X(M*N), WORK(*)
CHARACTER(1)           DIRECT
```

3 Description

C06PUF computes the two-dimensional discrete Fourier transform of a bivariate sequence of complex data values $z_{j_1 j_2}$, for $j_1 = 0, 1, \ldots, m-1$ and $j_2 = 0, 1, \ldots, n-1$.

The discrete Fourier transform is here defined by

$$\hat{z}_{k_1 k_2} = \frac{1}{\sqrt{mn}} \sum_{j_1=0}^{m-1} \sum_{j_2=0}^{n-1} z_{j_1 j_2} \times \exp\left(\pm 2\pi i \left(\frac{j_1 k_1}{m} + \frac{j_2 k_2}{n} \right) \right),$$

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 C06PUF 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 (1983) Self-sorting mixed-radix fast Fourier transforms *J. Comput. Phys.* **52** 1–23

5 Parameters

1: DIRECT – CHARACTER(1) *Input*

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

If the backward transform is to be computed then DIRECT must be set equal to 'B'.

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

2: M – INTEGER *Input*

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

Constraint: $M \geq 1$.

3: N – INTEGER *Input*

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

Constraint: $N \geq 1$.

4: X(M × N) – COMPLEX (KIND=nag_wp) array *Input/Output*

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

On exit: the corresponding elements of the computed transform.

5: WORK(∗) – COMPLEX (KIND=nag_wp) array *Workspace*

Note: the dimension of the array WORK must be at least $M \times N + N + M + 30$.

The workspace requirements as documented for C06PUF may be an overestimate in some implementations.

On exit: the real part of $WORK(1)$ 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. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

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.**

On exit: $IFAIL = 0$ unless the routine detects an error or a warning has been flagged (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 or warnings detected by the routine:

$IFAIL = 1$

On entry, $M < 1$.

$IFAIL = 2$

On entry, $N < 1$.

$IFAIL = 3$

On entry, DIRECT $\neq$ 'F' or 'B'.

$IFAIL = 4$

On entry, N has more than 30 prime factors.

IFAIL = 5

On entry, M has more than 30 prime factors.

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 is approximately proportional to $mn \times \log(mn)$, but also depends on the factorization of the individual dimensions m and n. C06PUF is faster if the only prime factors are 2, 3 or 5; and fastest of all if they are powers of 2.

9 Example

This example 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

```
!   C06PUF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module c06pufe_mod

!     C06PUF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: nin = 5, nout = 6
    Contains
      Subroutine readx(nin,x,n1,n2)
!       Read 2-dimensional complex data

!       .. Scalar Arguments ..
        Integer, Intent (In)                 :: n1, n2, nin
!       .. Array Arguments ..
        Complex (Kind=nag_wp), Intent (Out)  :: x(n1,n2)
!       .. Local Scalars ..
        Integer                              :: i, j
!       .. Executable Statements ..
        Do i = 1, n1
          Read (nin,*)(x(i,j),j=1,n2)
        End Do
        Return
      End Subroutine readx

      Subroutine writx(nout,x,n1,n2)
!       Print 2-dimensional complex data

!       .. Scalar Arguments ..
        Integer, Intent (In)                 :: n1, n2, nout
!       .. Array Arguments ..
```

```
        Complex (Kind=nag_wp), Intent (In)    :: x(n1,n2)
!       .. Local Scalars ..
        Integer                               :: i, j
!       .. Intrinsic Procedures ..
        Intrinsic                             :: aimag, real
!       .. Executable Statements ..
        Do i = 1, n1
          Write (nout,*)
          Write (nout,99999) 'Real ', (real(x(i,j)),j=1,n2)
          Write (nout,99999) 'Imag ', (aimag(x(i,j)),j=1,n2)
        End Do
        Return

99999   Format (1X,A,7F10.3/(6X,7F10.3))
      End Subroutine writx
    End Module c06pufe_mod

    Program c06pufe

!     C06PUF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: c06puf, nag_wp
      Use c06pufe_mod, Only: nin, nout, readx, writx
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Integer                               :: ieof, ifail, m, n
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable   :: work(:), x(:)
!     .. Executable Statements ..
      Write (nout,*) 'C06PUF Example Program Results'
!     Skip heading in data file
      Read (nin,*)
loop: Do
        Read (nin,*,Iostat=ieof) m, n
        If (ieof<0) Exit loop
        Allocate (work(m*n+n+m+30),x(m*n))
        Call readx(nin,x,m,n)
        Write (nout,*)
        Write (nout,*) 'Original data values'
        Call writx(nout,x,m,n)

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        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)
        Deallocate (x,work)

      End Do loop

    End Program c06pufe
```

9.2 Program Data

```
C06PUF Example Program Data
3  5                   : m, n
    ( 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)    : x
```

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
```