NAG Library Routine Document

C06PSF

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

C06PSF computes the discrete Fourier transforms of m sequences, stored as columns of an array, each containing n complex data values.

2 Specification

```
SUBROUTINE C06PSF (DIRECT, N, M, X, WORK, IFAIL)

INTEGER                N, M, IFAIL
COMPLEX (KIND=nag_wp) X(N*M), WORK(*)
CHARACTER(1)           DIRECT
```

3 Description

Given m sequences of n complex data values z_j^p, for $j = 0, 1, \ldots, n-1$ and $p = 1, 2, \ldots, m$, C06PSF simultaneously calculates the (**forward** or **backward**) discrete Fourier transforms of all the sequences defined by

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

(Note the scale factor $\frac{1}{\sqrt{n}}$ 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 C06PSF 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 (see Brigham (1974)) known as the Stockham self-sorting algorithm, which is described in Temperton (1983). Special code is provided for the factors 2, 3 and 5.

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: N – INTEGER *Input*

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

Constraint: $N \geq 1$.

3: M – INTEGER *Input*

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

Constraint: $M \geq 1$.

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

On entry: the complex data values z_j^p stored in $X((p-1) \times N + j + 1)$, for $j = 0, 1, \ldots, N-1$ and $p = 1, 2, \ldots, M$.

On exit: is overwritten by the complex transforms.

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

Note: the dimension of the array WORK must be at least $N \times M + N + 15$.

The workspace requirements as documented for C06PSF 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 $= 5$

An unexpected error has occurred in an internal call. Check all subroutine calls and array dimensions. Seek expert help.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

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 Parallelism and Performance

C06PSF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

C06PSF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken by C06PSF is approximately proportional to $nm\log(n)$, but also depends on the factors of n. C06PSF 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.

10 Example

This example reads in sequences of complex data values and prints their discrete Fourier transforms (as computed by C06PSF with DIRECT = 'F'). Inverse transforms are then calculated using C06PSF with DIRECT = 'B' and printed out, showing that the original sequences are restored.

10.1 Program Text

```
    Program c06psfe

!     C06PSF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: c06psf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ieof, ifail, j, m, n
```

```
!     .. Local Arrays ..
      Complex (Kind=nag_wp), Allocatable :: work(:), x(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: aimag, real
!     .. Executable Statements ..
      Write (nout,*) 'C06PSF 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(n*m+n+15),x(m*n))
        Do j = 1, m*n, n
          Read (nin,*)(x(j+i),i=0,n-1)
        End Do
        Write (nout,*)
        Write (nout,*) 'Original data values'
        Do j = 1, m*n, n
          Write (nout,*)
          Write (nout,99999) 'Real ', (real(x(j+i)),i=0,n-1)
          Write (nout,99999) 'Imag ', (aimag(x(j+i)),i=0,n-1)
        End Do

!       ifail: behaviour on error exit
!              =0 for hard exit, =1 for quiet-soft, =-1 for noisy-soft
        ifail = 0
        Call c06psf('F',n,m,x,work,ifail)

        Write (nout,*)
        Write (nout,*) 'Discrete Fourier transforms'
        Do j = 1, m*n, n
          Write (nout,*)
          Write (nout,99999) 'Real ', (real(x(j+i)),i=0,n-1)
          Write (nout,99999) 'Imag ', (aimag(x(j+i)),i=0,n-1)
        End Do

        Call c06psf('B',n,m,x,work,ifail)

        Write (nout,*)
        Write (nout,*) 'Original data as restored by inverse transform'
        Do j = 1, m*n, n
          Write (nout,*)
          Write (nout,99999) 'Real ', (real(x(j+i)),i=0,n-1)
          Write (nout,99999) 'Imag ', (aimag(x(j+i)),i=0,n-1)
        End Do
        Deallocate (x,work)
      End Do loop

99999 Format (1X,A,6F10.4)
    End Program c06psfe
```

10.2 Program Data

```
C06PSF Example Program Data
     3     6           : m, n
     (0.3854,0.5417)
     (0.6772,0.2983)
     (0.1138,0.1181)
     (0.6751,0.7255)
     (0.6362,0.8638)
     (0.1424,0.8723)
     (0.9172,0.9089)
     (0.0644,0.3118)
     (0.6037,0.3465)
     (0.6430,0.6198)
     (0.0428,0.2668)
     (0.4815,0.1614)
     (0.1156,0.6214)
```

```
   (0.0685,0.8681)
   (0.2060,0.7060)
   (0.8630,0.8652)
   (0.6967,0.9190)
   (0.2792,0.3355)   : x
```

10.3 Program Results

```
C06PSF Example Program Results

Original data values

Real     0.3854    0.6772    0.1138    0.6751    0.6362    0.1424
Imag     0.5417    0.2983    0.1181    0.7255    0.8638    0.8723

Real     0.9172    0.0644    0.6037    0.6430    0.0428    0.4815
Imag     0.9089    0.3118    0.3465    0.6198    0.2668    0.1614

Real     0.1156    0.0685    0.2060    0.8630    0.6967    0.2792
Imag     0.6214    0.8681    0.7060    0.8652    0.9190    0.3355

Discrete Fourier transforms

Real     1.0737   -0.5706    0.1733   -0.1467    0.0518    0.3625
Imag     1.3961   -0.0409   -0.2958   -0.1521    0.4517   -0.0321

Real     1.1237    0.1728    0.4185    0.1530    0.3686    0.0101
Imag     1.0677    0.0386    0.7481    0.1752    0.0565    0.1403

Real     0.9100   -0.3054    0.4079   -0.0785   -0.1193   -0.5314
Imag     1.7617    0.0624   -0.0695    0.0725    0.1285   -0.4335

Original data as restored by inverse transform

Real     0.3854    0.6772    0.1138    0.6751    0.6362    0.1424
Imag     0.5417    0.2983    0.1181    0.7255    0.8638    0.8723

Real     0.9172    0.0644    0.6037    0.6430    0.0428    0.4815
Imag     0.9089    0.3118    0.3465    0.6198    0.2668    0.1614

Real     0.1156    0.0685    0.2060    0.8630    0.6967    0.2792
Imag     0.6214    0.8681    0.7060    0.8652    0.9190    0.3355
```