

NAG Library Routine Document

C06HAF

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

C06HAF computes the discrete Fourier sine transforms of m sequences of real data values. This routine is designed to be particularly efficient on vector processors.

2 Specification

```
SUBROUTINE C06HAF(M, N, X, INIT, TRIG, WORK, IFAIL)
INTEGER          M, N, IFAIL
double precision X(M*N), TRIG(2*N), WORK(M*N)
CHARACTER*1      INIT
```

3 Description

Given m sequences of $n - 1$ real data values x_j^p , for $j = 1, 2, \dots, n - 1$ and $p = 1, 2, \dots, m$, C06HAF simultaneously calculates the Fourier sine transforms of all the sequences defined by

$$\hat{x}_k^p = \sqrt{\frac{2}{n}} \sum_{j=1}^{n-1} x_j^p \times \sin\left(jk\frac{\pi}{n}\right), \quad k = 1, 2, \dots, n - 1 \text{ and } p = 1, 2, \dots, m.$$

(Note the scale factor $\sqrt{\frac{2}{n}}$ in this definition.)

The Fourier sine transform defined above is its own inverse, and two consecutive calls of this routine with the same data will restore the original data.

The transform calculated by this routine can be used to solve Poisson's equation when the solution is specified at both left and right boundaries (see Swarztrauber (1977)). (See the C06 Chapter Introduction.)

The routine uses a variant of the fast Fourier transform (FFT) algorithm (see Brigham (1974)) known as the Stockham self-sorting algorithm, described in Temperton (1983), together with pre- and post-processing stages described in Swarztrauber (1982). Special coding is provided for the factors 2, 3, 4, 5 and 6. This routine is designed to be particularly efficient on vector processors, and it becomes especially fast as m , the number of transforms to be computed in parallel, increases.

4 References

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

Swarztrauber P N (1977) The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle *SIAM Rev.* **19** (3) 490–501

Swarztrauber P N (1982) Vectorizing the FFT's *Parallel Computation* (ed G Rodrigue) 51–83 Academic Press

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

5 Parameters

- 1: M – INTEGER Input

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

Constraint: $M \geq 1$.

- 2: N – INTEGER Input

On entry: one more than the number of real values in each sequence, i.e., the number of values in each sequence is $n - 1$.

Constraint: $N \geq 1$.

- 3: X(M × N) – **double precision** array Input/Output

On entry: the data must be stored in X as if in a two-dimensional array of dimension (1 : M, 1 : N); each of the m sequences is stored in a **row** of the array. In other words, if the $n - 1$ data values of the p th sequence to be transformed are denoted by x_j^p , for $j = 1, 2, \dots, n - 1$ and $p = 1, 2, \dots, m$, then the first $m(n - 1)$ elements of the array X must contain the values

$$x_1^1, x_1^2, \dots, x_1^m, x_2^1, x_2^2, \dots, x_2^m, \dots, x_{n-1}^1, x_{n-1}^2, \dots, x_{n-1}^m.$$

The n th element of each row x_n^p , for $p = 1, 2, \dots, m$, is required as workspace. These m elements may contain arbitrary values on entry, and are set to zero by the routine.

On exit: the m Fourier transforms stored as if in a two-dimensional array of dimension (1 : M, 1 : N). Each of the m transforms is stored in a **row** of the array, overwriting the corresponding original sequence. If the $n - 1$ components of the p th Fourier sine transform are denoted by $\hat{x}_k^p$, for $k = 1, 2, \dots, n - 1$ and $p = 1, 2, \dots, m$, then the mn elements of the array X contain the values

$$\hat{x}_1^1, \hat{x}_1^2, \dots, \hat{x}_1^m, \hat{x}_2^1, \hat{x}_2^2, \dots, \hat{x}_2^m, \dots, \hat{x}_{n-1}^1, \hat{x}_{n-1}^2, \dots, \hat{x}_{n-1}^m, 0, 0, \dots, 0 \text{ (} m \text{ times)}.$$

If $n = 1$, the m elements of X are set to zero.

- 4: INIT – CHARACTER*1 Input

On entry: indicates whether trigonometric coefficients are to be calculated.

INIT = 'I'

Calculate the required trigonometric coefficients for the given value of n , and store in the array TRIG.

INIT = 'S' or 'R'

The required trigonometric coefficients are assumed to have been calculated and stored in the array TRIG in a prior call to one of C06HAF, C06HBF, C06HCF or C06HDF. The routine performs a simple check that the current value of n is consistent with the values stored in TRIG.

Constraint: INIT = 'I', 'S' or 'R'.

- 5: TRIG(2 × N) – **double precision** array Input/Output

On entry: if INIT = 'S' or 'R', TRIG must contain the required trigonometric coefficients calculated in a previous call of the routine. Otherwise TRIG need not be set.

On exit: contains the required coefficients (computed by the routine if INIT = 'I').

- 6: WORK(M × N) – **double precision** array Workspace

- 7: 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.

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, 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.**

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, INIT $\neq$ 'T', 'S' or 'R'.

IFAIL = 4

Not used at this Mark.

IFAIL = 5

On entry, INIT = 'S' or 'R', but the array TRIG and the current value of N are inconsistent.

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 by C06HAF is approximately proportional to $nm \log n$, but also depends on the factors of n . C06HAF 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 example reads in sequences of real data values and prints their Fourier sine transforms (as computed by C06HAF). It then calls C06HAF again and prints the results which may be compared with the original sequence.

9.1 Program Text

```

*      C06HAF Example Program Text
*      Mark 14 Revised. NAG Copyright 1989.
*      .. 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 ..
      DOUBLE PRECISION TRIG(2*NMAX), WORK(MMAX*NMAX), X(NMAX*MMAX)
*      .. External Subroutines ..
      EXTERNAL         C06HAF
*      .. Executable Statements ..
      WRITE (NOUT,*) 'C06HAF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
20     READ (NIN,*,END=120) M, N
      IF (M.LE.MMAX .AND. N.LE.NMAX) THEN
        DO 40 J = 1, M
          READ (NIN,*) (X((I-1)*M+J),I=1,N-1)
40      CONTINUE
        WRITE (NOUT,*)
        WRITE (NOUT,*) 'Original data values'
        WRITE (NOUT,*)
        DO 60 J = 1, M
          WRITE (NOUT,99999) (X((I-1)*M+J),I=1,N-1)
60      CONTINUE
        IFAIL = 1
*
*      -- Compute transform
      CALL C06HAF(M,N,X,'Initial',TRIG,WORK,IFAIL)
*
      IF (IFAIL.EQ.0) THEN
        WRITE (NOUT,*)
        WRITE (NOUT,*) 'Discrete Fourier sine transforms'
        WRITE (NOUT,*)
        DO 80 J = 1, M
          WRITE (NOUT,99999) (X((I-1)*M+J),I=1,N-1)
80      CONTINUE
*
*      -- Compute inverse transform
      CALL C06HAF(M,N,X,'Subsequent',TRIG,WORK,IFAIL)
*
        WRITE (NOUT,*)
        WRITE (NOUT,*)
        +      'Original data as restored by inverse transform'
        WRITE (NOUT,*)
        DO 100 J = 1, M
          WRITE (NOUT,99999) (X((I-1)*M+J),I=1,N-1)
100     CONTINUE
        GO TO 20
      ELSE
        WRITE (NOUT,*)
        WRITE (NOUT,99998) ' ** C06HAF returned with IFAIL = ',
        +      IFAIL
        END IF
      ELSE
        WRITE (NOUT,*) 'Invalid value of M or N'
      END IF
120  CONTINUE
*
99999 FORMAT (6X,6F10.4)
99998 FORMAT (1X,A,I5)
      END

```

9.2 Program Data

C06HAF Example Program Data

```
3 6 : Number of sequences, M, (number of values in each sequence)+1, N
0.6772 0.1138 0.6751 0.6362 0.1424 : X, sequence 1
0.2983 0.1181 0.7255 0.8638 0.8723 : X, sequence 2
0.0644 0.6037 0.6430 0.0428 0.4815 : X, sequence 3
```

9.3 Program Results

C06HAF Example Program Results

Original data values

0.6772	0.1138	0.6751	0.6362	0.1424
0.2983	0.1181	0.7255	0.8638	0.8723
0.0644	0.6037	0.6430	0.0428	0.4815

Discrete Fourier sine transforms

1.0014	0.0062	0.0834	0.5286	0.2514
1.2477	-0.6599	0.2570	0.0858	0.2658
0.8521	0.0719	-0.0561	-0.4890	0.2056

Original data as restored by inverse transform

0.6772	0.1138	0.6751	0.6362	0.1424
0.2983	0.1181	0.7255	0.8638	0.8723
0.0644	0.6037	0.6430	0.0428	0.4815
