

10.5 Primitive Fast Fourier Transform

A. Purpose

This subroutine is not intended to be used directly in applications. It is used by other library subroutines and is described here to supplement the functional descriptions of these other routines. This subroutine computes one-dimensional complex Fourier transforms using the Cooley-Tukey **Fast Fourier Transform (FFT)**. Complex values z and complex Fourier coefficients ζ are related by

$$z_j = \sum_{k=0}^{N-1} \zeta_k W^{jk}, \quad j = 0, 1, \dots, N-1, \quad (1)$$

where $N = 2^M$, $W = e^{2\pi i/N} = \cos 2\pi/N + i \sin 2\pi/N$, and $i^2 = -1$. This subroutine is used by the library subroutines:

SRFT1 One-dimensional **Real Fourier Transform**
SCFT **Complex Fourier Transform**
SSCT **Sine, Cosine, or Trigonometric Transform**
SRFT **Real Fourier Transform**

The last three of these do transformations in up to 6 dimensions. For any of the above tasks, one should refer to the corresponding chapter.

B. Usage

B.1 Program Prototype, Evaluate a Fourier Series, Single Precision

REAL A(\geq IR - 1 + KS \times 2^{MM}), S(\geq NT - 1)
INTEGER MT, NT, MM, KS, ILAST, KE(30)
LOGICAL NEEDST
COMMON / CSFFTC / Contains **NEEDST**, **MT**,
NT, **MM**, **KS**, **ILAST**, **KE**

(Contents and structure of the common block are subject to change.)

If the sine table S has not been computed, set NEEDST=.TRUE., and

```
CALL SFFT(A(IR), A(II), S)
```

To evaluate a complex Fourier series, assign values to IR (the index of the real part of the first coefficient, ζ , in the real array A()), II (the index of the imaginary part of the first coefficient, ζ , in the real array A()), the Fourier coefficients, ζ , in A(), and values in the common block.

```
CALL SFFT(A(IR), A(II), S)
```

On return the array A() contains the values z , and S() contains the sine table.

©1997 Calif. Inst. of Technology, 2015 Math à la Carte, Inc.

B.2 Argument and Common Definitions

A() [inout] Real array holding the complex coefficients.

S() [inout] Array used to hold the sine table. If NEEDST is false on entry, then S should have been computed on a previous entry to SFFT.

NEEDST [inout] If true the sine table S is computed and no transform is computed. NEEDST is always false on exit.

MT [in] base 2 log of NT.

NT [inout] Number of entries in the sine table + 1. If NEEDST is false, then NT should have been computed on a previous entry to SFFT.

MM [in] \log_2 (number of complex Fourier coefficients).

KS [in] distance in memory between successive entries in A. The complex coefficients used in the computation are $A(\text{IR} + I \times \text{KS}) + i A(\text{II} + I \times \text{KS})$, $I = 0, 1, \dots, 2^{\text{MM}-1}$, where $i^2 = -1$.

ILAST [in] = $\text{KS} \times 2^{\text{MM}}$.

KE() [in] array with $\text{KE}(L) = \text{KS} \times 2^{\text{MM}-L}$.

B.3 Modifications to Obtain Coefficients Given Values of the Series

The complex values $z \times 2^{-\text{MM}}$ are stored in A(), and then

```
CALL SFFT(A(II), A(IR), S)
```

On return the array A() contains the complex Fourier Coefficients ζ .

B.4 Modifications for Double Precision

Change SFFT to DFFT, the REAL type statement to DOUBLE PRECISION, and the name of the common block to CDFFTC.

C. Examples and Remarks

Examples of use can be found in source listings for the library subprograms **SRFT1**, **SCFT**, **SSCT**, and **SRFT**.

D. Functional Description

This subprogram uses the Cooley-Tukey algorithm [1] to evaluate series of the form given in Eq. (1) above. The radix (4 + 2) algorithm is used, which means that N is factored as $4^q 2^r$, $r = 0$ or 1. The algorithm involves q radix 4 stages, and if M is odd, one radix 2 stage. The input array is scrambled based on the KE() before the

first stage so that results come out in the correct order. Details can be found in [2].

The procedure for computing ζ_k given z_j involves dividing by N , and interchanging the roles of the real and imaginary parts for both the input and output arrays. Since $i\bar{z} = \Im z + i\Re z$ (\bar{z} = conjugate of z), it follows that using this procedure is equivalent to writing

$$i\bar{\zeta}_k = \frac{1}{N} \sum_{j=0}^{N-1} \bar{z}_j W^{jk}, \quad k = 0, 1, \dots, N-1,$$

which in turn can be verified by dividing both sides by i , conjugating both sides ($\overline{\bar{W}} = W^{-1}$), and substituting the expression for z_j given by Eq. (1), yielding

$$\begin{aligned} \zeta_K &= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} \zeta_k W^{jk} W^{-jK} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \zeta_k \sum_{j=0}^{N-1} W^{j(k-K)} = \zeta_K \end{aligned}$$

since

$$\sum_{j=0}^{N-1} W^{jL} = \begin{cases} N & \text{if } L = 0 \\ \frac{1 - W^{LN}}{1 - W^L} = 0 & \text{if } L = \pm 1, \dots, \pm(N-1). \end{cases}$$

References

1. J. W. Cooley and J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, **Math. of Comp.** **19** (1965) 297–301.
2. Fred T. Krogh, **FFT — Primitive Fast Fourier Transform**. TU Doc. CP-2311, NPO 11650, Jet Propulsion Laboratory, Pasadena, CA (1970).

E. Error Procedures and Restrictions

No tests are made for bad argument values, e.g., $MM < 0$.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry	Required Files
DFFT	DFFT
SFFT	SFFT