AN ALGORITHM FOR THE ESTIMATION OF A FOURIER TRANSFORM

Z. T. Volkova and S. I. Sirvidas

UDC 681.3.06:51

The great labor-consuming character of the calculation of the estimates of a Fourier transform in relation to time as well as the memory of digital computers hampers the use of the Fourier transform in the investigations realizable on a digital computer. For the calculation of the estimates of a Fourier transform on a digital computer by the direct method we need at least $4N^2$ operations of multiplication where $N$ determines the size of the discrete time sequence.

An algorithm economical in relation to the time of calculation of the estimates of a Fourier transform was first given in 1965 by Cooley and Tukey [1] and was called the method of quick Fourier transform in the literature. The quick Fourier transform is a highly effective procedure for the calculation of a discrete Fourier transform. The algorithm allows us to calculate iteratively the estimates of a Fourier transform in $n$ steps ($N = 2^n$), which shortens the amount of calculations by $2N/\log_2 N$ times.

The realization of the algorithm of Cooley and Tukey on a universal digital computer is made difficult because of the necessity to introduce special commands of the type "inversion of code" or a standard program realizing the so-called "dual inversion" [2].

In the present article we have posed the problem of developing a rational algorithm which allows us to obtain the estimates of a Fourier transform on a universal digital computer which is effective in relation to the amount of calculation and memory, and which does not require the use of special commands.

**Principle of Construction of the Algorithm**

If a set of $N$ complex numbers representing the values of the function $X$ at the moments of time $t = 0$, $\Delta t$, $2\Delta t$, $\ldots$, $(N-1)\Delta t$ ($X_0$, $X_1$, $X_2$, $\ldots$, $X_r$, $\ldots$, $X_{N-1}$), is given, then the estimates for its Fourier transform are given by the formula

$$S_u = \sum_{r=0}^{N-1} X_r e^{i2\pi ur/N}, \quad u = 0, 1, 2, \ldots, N - 1,$$

(1)

where

$$q = \exp\left(-i\frac{2\pi u}{N}\right).$$

(2)

For $N = 2^n$ we can calculate $S_u$ in $n$ steps by using the following recurrence relation [1]:

$$X_r^{(i)} = X_r^{(i-1)} + q^{r/2} X_{r-2^{i-1}}^{(i-1)} e^{-2\pi i u/2^i},$$

$$r = 2^{i-1} 2^{i-1} + 1, \ldots, 2^{i-1} - 1,$$

$$X_0^{(i)} = X_0, \quad \alpha = 0, 1, \ldots, N - 1,$$

(3)

where \( \mu_r^{(l)} = \frac{N}{2^l} \cdot S_u = \sum_{(a)} \mu_r \), \( u = 0, 1, \ldots, N-1 \), and \( \delta (u) \) is the inversion of the dual expression of the number \( u \) \([1]\). We shall call the set of numbers \( \{X_r^{(l)}\} \) the \( l \)th intermediate array. The set of the numbers of the array which are formed by (3) from the numbers of the \((l-1)\)th array with the same \( X_r^{(l)} \) will be called a subarray of it. The number \( \beta \) of the subarrays of the \( l \)th array is given by \( \beta = 2^l \), \( l = 0, 1, \ldots, n \). Thus, the number of subarrays in the \( l \)th array is twice as much as that in the \((l-1)\)th array. The size of a subarray is determined by the number \( \alpha \): \[
\alpha = \frac{N}{2^l}, \quad \alpha = N, \frac{N}{2}, \ldots, 1.
\]

The exponents of the numbers of the subarrays \((k_l = 1 - \beta / 2)\) in the first half of the \( l \)th array coincide with the exponents of the numbers of the \((l-1)\)th array. The exponents of all the numbers of the subsequent subarrays \((k_A = \beta / 2 + 1 - \beta)\) of the \( l \)th array are formed from the sum of the exponents of the first half of the \( l \)th array and the term determined by the size of the array and the number of the formed array: \[
\mu_{s(l)} = \mu_s + \frac{N}{2^l},
\]
or \[
\mu_B = \mu_B + \alpha. \quad (4)
\]

For \( l = 0, \beta = 1 \), we have \( \mu_1 = 0 \); for \( l = 1, \beta = 2 \), we have \( \mu_2 = 1 \); for \( l = 2, \beta = 4 \), we have \( \mu_{k1} = \mu_0 = 0 \); for \( l = 3, \beta = 8 \), we have \( \mu_{k2} = \mu_1 + 0 + \frac{N}{2} \); for \( l = 4, \beta = 16 \), we have \( \mu_{k3} = \mu_2 + 0 + \frac{N}{2} + \frac{N}{4} \), and so on.