RECEIVED BY DTIC NOV 16 1971

MASTER

ON-LINE FAST FOURIER TRANSFORMS (a)
ON A HYBRID COMPUTER

by

P. J. Dionne

and

C. R. Cole

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

SYSTEMS ENGINEERING
COMPUTERS AND CONTROL SECTION

October 11, 1971

Battelle Memorial Institute
Pacific Northwest Laboratories
Richland, WA 99352

(a) This paper is based on work performed under United States Atomic Energy Commission Contract AT(45-1)-1830.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

ON-LINE FAST FOURIER TRANSFORMS ON A HYBRID COMPUTER

Introduction

Historically, Direct Fourier Transforms (DFT) have been applied on large digital machines. There is evidence to show that some of these codes used hundreds of hours of computer time and required N^2 operations to compute Fourier transforms of N data points. A recent paper by Cooley⁽¹⁾ and co-workers describes a Fast Fourier Transform (FFT) technique which incorporates a successive doubling algorithm. Their FFT algorithm requires only $N \log(N)$ operations to compute a Fourier transform on N data points. Cooley lists a FORTRAN subroutine of the algorithm in his paper. It is this subroutine which was recoded for the hybrid computer. A more recent paper by Cooley⁽²⁾ shows how the algorithm may be applied using only 50% of the core space required by the original code.

Theory

Brigham and Morrow⁽³⁾ treat the basic theory in detail. Briefly

$$S(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt$$

where $x(t)$ is a continuous aperiodic function of time and $S(f)$ is commonly referred to as the Fourier transform of $x(t)$. Fourier transforms are useful in a number of applications. In particular, the solution of differential equations is simplified because the transform converts a differential equation with time as the independent variable into an algebraic equation with f as the independent variable. The transform provides an easy method for relating the input and output variables of linear time-invariant systems encountered in electrical, mechanical, and optical systems. The Fourier transform yields a complex function defined in a one-dimensional frequency space; consequently, the inverse transformation is easily adapted to numerical methods. The Fourier transform is an excellent tool for analyzing physical data because numerical methods and digital computers are generally used for this purpose.

One Fourier-transform property having application in the analysis of linear time-invariant systems is that the transform of the convolution of two functions is equal to the product of the transforms of the individual functions. Another application of the Fourier transform is the calculation of power spectra. A power spectrum is defined as a Fourier transformation of the autocorrelation

-2-

function. Because the Fourier transform is indeed a powerful tool of analysis, it is not surprising that a search was begun to establish techniques to compute the Fourier transform numerically. The discrete Fourier transform therefore evolved, but it was considered impractical until the development of the fast Fourier transform.

The fast Fourier transform method is simply an algorithm for reducing the number of calculations necessary for evaluating the discrete Fourier transform $S(f)$ of a sampled time function $x(t)$. For a sampled function with N sampled data points Δt time increments apart, we have

$$S(f_n) = \Delta t \sum_{k=0}^{N-1} x(t_k) e^{-j2\pi f_n t_k} \quad (2)$$

$$n = 0, \pm 1, \dots, \pm N/2$$

$$\text{and } x(t_k) = \Delta f \sum_{n=N/2}^{N/2} S(f_n) e^{j2\pi f_n t_k} \quad (3)$$

$$k = 0, 1, \dots, N-1$$

Now if we let $t_k = k\Delta t$ and $f_n = n\Delta f$ where $\Delta t = T/N$ and $\Delta f = 1/T$ and T is the length of sample and N is the total number of points then

$$S(n) = \Delta t \sum_{k=0}^{N-1} x(k) e^{-j2\pi (nk)/N} \quad (4)$$

$$n = 0, 1, \dots, N-1$$

$$\text{and } x(k) = \Delta f \sum_{n=0}^{N-1} S(n) e^{j2\pi (nk)/N} \quad (5)$$

$$k = 0, 1, \dots, N-1$$

This last change, or substitution, is made to simplify computational procedure. The $n = N/2$ term corresponds to the Nyquist folding frequency. The FFT method is simply concerned with evaluation of (4). If we let $W^{nk} = \exp(j2\pi nk/N)$ then (4) becomes (6) in matrix form:

-3-

$$[S(n)] = [W^{nk}] [X_0(k)] \quad (6)$$

Expanded Sample for $N = 4$

$$\begin{bmatrix} S(0) \\ S(1) \\ S(2) \\ S(3) \end{bmatrix} = \begin{bmatrix} W^{00} & W^{01} & W^{02} & W^{03} \\ W^{10} & W^{11} & W^{12} & W^{13} \\ W^{20} & W^{21} & W^{22} & W^{23} \\ W^{30} & W^{31} & W^{32} & W^{33} \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \quad (7)$$

From the matrix form of (6), it can be seen that the number of operations necessary to evaluate the S 's is N^2 complex additions and multiplications. The FFT algorithm reduces this for N sample points (where $N = 2^Y$) to YN complex operations. The calculation redundancies of the direct method for evaluating (6) are reduced in three ways: the first is due to the periodicity of W^{nk} (i.e., $W^{nk} = W^{nk \text{ Mod } (N)}$) and since $W^{00} = 1$, $\therefore$ (7) becomes (8).

$$\begin{bmatrix} S(0) \\ S(1) \\ S(2) \\ S(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & 1 & W^2 \\ 1 & W^3 & W^2 & 1 \end{bmatrix} \begin{bmatrix} X_0(0) \\ X_0(1) \\ X_0(2) \\ X_0(3) \end{bmatrix} \quad (8)$$

The second reduction occurs due to a factoring of the $[W]$ matrix into a series of product matrices in such a manner that each of the new factored matrices has the property of minimizing the number of complex multiplications and additions. The third reduction in the number of calculations occurs only if $N = 2^M$ (where M is an integer); the reduction in complex operations occurs by avoiding multiplications when the powers of W_N are simple numbers like ± 1 or $\pm i$.

Experimental Protocol

The design lifetime of the fuel pins in the IMFBR program is 9000 hours. Argonne National Laboratories has been involved in a study of the effect of liquid sodium on the fuel pins. Argonne had some fuel pins in a liquid sodium bath for a period of 6000 hours or 2/3 of the design lifetime. An effort was made to provide a realistic environment for the fuel pins, excluding irradiation. Upon removal from the sodium bath, the fuel pins were to be shipped from Argonne to HEDL for structural evaluation. Concern was expressed that shipping damage to the fuel pins might override the effects of the sodium bath.

-4-

BNW helped WADCO conduct a study to determine the best mode of shipping. A moving van was secured and a dummy fuel element was crated and placed in several positions within the van. Accelerometer data were recorded on magnetic tape. The data indicated that the best shipping position was on the bed of the van with the fuel pin axis pointing front to back. Then several runs were made with the fuel pin box strapped to the floor, or mounted on foam cushions and secured with steel or nylon straps. A certain type of foam cushion and nylon straps were chosen as the best method of shipping the fuel element. Three, three-coordinate accelerometers were used to transmit information during the shipment. One was placed on the floor and two on the shipping crate, one at each end. A magnetic tape recorder was used to record the analog signals. Approximately sixteen hours of data were gathered during the test runs and the trip from Argonne to HEDL.

Due to transcribing the data from one recorder to another and due to difficulties in gathering data on the trip, only three usable signals and a voice channel were received for analysis. The signals were T1, the transverse acceleration from the forward accelerometer, and L2 and V2, the lateral and vertical signals from the rear accelerometer.

During playback, the data were reproduced in real time at 1 7/8 ips. Since the accelerometer signal was attenuated by twenty before recording, the signal conditioning circuit multiplies the incoming signal by twenty. This restores the conversion of fifteen volts being equivalent to 1 g acceleration. The rest of the signal conditioning circuit is used for gain and bias adjustments. At the time of gathering data, a 100 Hz 10v_{pp} sinewave was recorded for calibration purposes. The calibration circuit eliminates gain and bias differences between recorders.

The spectral analysis was performed in three steps: (1) sampling the analog signals and storing the digitized values on DECTape, (2) an FFT analysis, and (3) a display of the results.

The FFT algorithm was implemented on the BNW hybrid computer. It was checked out using square pulses and sinewaves whose Fourier transforms are known. The analysis was applied to two sets of field data. The first was accelerometer data and the second was wind velocity data. The program is versatile since it

-5-

can sample very low frequencies up to frequencies approaching 4000 Hz. If, for some reason, analog data with frequency components up to 120 kHz require analysis, then the recorded data may be played back at slower tape speeds. The range of frequency to be analyzed may be selected according to the formulas:

$$\begin{aligned} T \cdot N &= 1/\Delta f \\ f \cdot N &= 1/\Delta t \end{aligned} \tag{9}$$

where T is the length of the sample

N is the number of samples (where $N = 2^m$ and m is an integer)

f is the maximum frequency

Δf is the frequency difference between samples

Δt is the time difference between samples.

The sampling program was designed around the size of the PDP-7 core memory and the number of signals to be sampled. The core size is 8192 words and, originally, six signals from two accelerometers were to be sampled. N was chosen to be $2^{10} = 1024$. Six times 1024 is 6144 words leaving approximately 2000 words for the program and bootstrap. The accelerometer frequency bandwidth was 150 Hz so 200 Hz was arbitrarily chosen as the spectrum to be analyzed. Applying Equation (9) yielded a $\Delta f \approx .2$ Hz and $T \approx 5$ seconds. An almost infinite variety of frequency spectra may be chosen by changing Δt . For high frequencies, Δt will be small; for low frequencies, Δt will be large. Another way of increasing the frequency spectrum is to increase N. If N were doubled to 2048, then only three signals could be sampled simultaneously.

When the sampling program has completed sampling the data and storing the digitized data on DEC tape, the analysis program is automatically called into core and it begins analyzing the data one signal at a time; that is, the first 1024 words, then the second 1024 words, and so on until all six data arrays have been analyzed. After each analysis, the sampled data and its power spectra are printed out in graphical form on the Gould terminal.

Using fixed point arithmetic, which has limited accuracy, the analysis of the 1024 word matrix takes 2 seconds; using the more accurate floating point package, the analysis takes 11 seconds. The analysis program is designed so that the frequency spectrum of 200 Hz covers -100 Hz to +100 Hz with the power spectra of the plus frequencies. Therefore, the printout yields only the power spectra from 0-100 Hz. The algorithm was designed to conserve core space so the results wind up in the original 1024 data locations.

Results

Samples of the printout are shown in Figures 2-6. We show the results of a fixed point analysis, and a floating point analysis of a pulse in Figure 4 a-c. The fixed point analysis is displayed both linearly and logarithmically. The vertical axes are normalized to the highest non-zero frequency component. To demonstrate that the program does indeed calculate valid Fourier transforms, a series of square pulses that become longer in duration is shown in Figure 5. The first pulse is representative of a delta function which has a flat spectrum.

Figure 6 shows the FFT algorithm applied to two types of real world problems. The first is a sample of the accelerometer data while the second is a sample of wind velocity data. Wind velocity spectra is normally in the very low frequency range of .01-1 Hz so the tape was speeded up by 16 for playback into the sampling program.

References

1. Cooley, James W., Peter A. W. Lewis, and Peter D. Welch, "The Fast Fourier Transform and Its Applications," *IEEE Transactions on Education*, Vol. 12, No. 1, March 1969.
2. Cooley, J.W., P. A. Lewis and P. D. Welch, "The Fast Fourier Transform Algorithm: Programming Considerations in the Calculation of Sine, Cosine, and Laplace Transforms," *Journal of Sound Vibration*, Vol. 12, No. 3, pp. 315-337, 1970.
3. Brigham, E.O. and K. E. Morrow, "The Fast Fourier Transform," *IEEE Spectrum*, December 1967.