Reprinted from *AFIPS Proc. 1966 Spring Joint Computer Conf.*, Vol. 28, Spartan Books, 1966, pp. 229–233

HIGH-SPEED CONVOLUTION AND CORRELATION*

Thomas G. Stockham, Jr.
Massachusetts Institute of Technology, Project MAC
Cambridge, Massachusetts

INTRODUCTION

Cooley and Tukey[1] have disclosed a procedure for synthesizing and analyzing Fourier series for discrete periodic complex functions.† For functions of period N, where N is a power of 2, computation times are proportional to $N \log_2 N$ as expressed in Eq. (0).

$$T_{ct} = k_{ct} N \log_2 N \tag{0}$$

where k_{ct} is the constant of proportionality. For one realization for the IBM 7094, k_{ct} has been measured at 60 μsec. Normally the times required are proportional to N^2. For $N = 1000$ speed-up factors in the order of 50 have been realized! Eq. (1b) synthesizes the Fourier series in question. The complex Fourier coefficients are given by the analysis equation, Eq. (1a).

$$F(k) = \sum_{j=0}^{N-1} f(j) w^{-jk} \tag{1a}$$

$$f(j) = \frac{1}{N} \sum_{k=0}^{N-1} F(k) w^{jk} \tag{1b}$$

where $w = e^{2\pi i/N}$, the principal Nth root of unity. The functions f and F are said to form a discrete periodic complex transform pair. Both functions are of period N since

$$F(k) = F(k + cN) \tag{2a}$$

and

$$f(j) = f(j + cN) \tag{2b}$$

TRANSFORM PRODUCTS

Consider two functions g and h and their transforms G and H. Let G and H be multiplied to form the function C according to Eq. (3),

$$C(k) = G(k) \times H(k) \tag{3}$$

and consider the inverse transform $c(j)$. $c(j)$ is given by Eq. (4)

$$\begin{aligned} c(j) &= \frac{1}{N} \sum_{J=0}^{N-1} g(J) h(j - J) \\ &= \frac{1}{N} \sum_{J=0}^{N-1} h(J) g(j - J) \end{aligned} \tag{4}$$

as a sum of lagged products where the lags are performed circularly. Those values that are shifted from one end of the summation interval are circulated into the other.

The time required to compute $c(j)$ from either form of Eq. (4) is proportional to N^2. If one computes the transforms of g and h, performs the multiplication of Eq. (3), and then computes the inverse

*Work reported herein was supported (in part) by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01).

†To be able to use this procedure the period must be a highly composite number.

transform of C, one requires a time given by Eq. (5)

$$T_{circ} = 3\,k_{ct}\,N\log_2 N + k_m N = k_{circ}\,N(\log_2 N + \mu) \quad (5)$$

where $k_{circ} = 3k_{ct}$, $\mu = k_m/k_{circ}$, and $k_m N$ is the time required to compute Eq. (3). Of course this assumes N is a power of 2. Similar savings would be possible provided N is a highly composite number.

APERIODIC CONVOLUTION

The circular lagged product discussed above can be alternately regarded as a convolution of periodic functions of equal period. Through suitable modification a periodic convolution can be used to compute an aperiodic convolution when each aperiodic function has zero value everywhere outside some single finite aperture.

Let the functions be called $d(j)$ and $s(j)$. Let the larger finite aperture contain M discrete points and let the smaller contain N discrete points. The result of convolving these functions can be obtained from the result of circularly convolving suitable augmented functions. Let these augmented functions be periodic of period L, where L is the smallest power of 2 greater than or equal to $M + N$. Let them be called $da(j)$ and $sa(j)$ respectively, and be formed as indicated by Eq. (6).

$$\begin{aligned} fa(j) &= f(j + j_0) && 0 \le j \le M - 1 \\ &= 0 && M \le j \le L - 1 \\ &= fa(j + nL) && \text{otherwise} \end{aligned} \quad (6)$$

where j_0 symbolizes the first point in the aperture of the function in question. The intervals of zero values permit the two functions to be totally non-overlapped for at least one lagged product even though the lag is a circular one. Thus, while the result is itself a periodic function, each period is an exact replica of the desired aperiodic result.

The time required to compute this result is given in Eq. (7).

$$T_{aper} = k_{circ}\,L(\log_2 L + \mu) \quad (7)$$

where $M + N \le L < 2(M + N)$. For this case, while L must be adjusted to a power of 2 so that the high-speed Fourier transform can be applied, no restrictions are placed upon the values of either M or N.

SECTIONING

Let us assume that M is the aperture of $d(j)$ and N is that of $s(j)$. In situations where M is considerably larger than N, the procedure may be further streamlined by sectioning $d(j)$ into pieces each of which contains P discrete points where $P + N = L$, a power of 2. We require K sections where

$$K = \text{least integer} \ge M/P \quad (8)$$

Let the ith section of $d(j)$ be called $d_i(j)$. Each section is convolved aperiodically with $s(j)$ according to the discussion of the previous section, through the periodic convolution of the augmented sections, $da_i(j)$ and $sa(j)$.

Each result section, $r_i(j)$, has length $L = P + N$ and must be additively overlapped with its neighbors to form the composite result $r(j)$ which will be of length

$$KP + N \ge M + N \quad (9a)$$

If $r_i(j)$ is regarded as an aperiodic function with zero value for arguments outside the range $0 \le j \le L - 1$, these overlapped additions may be expressed as

$$r(j) = \sum_{i=0}^{K-1} r_i(j - iP) \quad j = 0,1,\ldots KP + N - 1 \quad (9b)$$

Each overlap margin has width N and there are $K - 1$ of them.

The time required for this aperiodic sectioned convolution is given in Eq. (10).

$$\begin{aligned} T_{sect} &= k_{ct}(P + N)\log_2(P + N) \\ &\quad + 2Kk_{ct}(P + N)\log_2(P + N) \\ &\quad + Kk_{aux}(P + N) \\ &= k_{ct}(2K + 1)(P + N)\log_2(P + N) \\ &\quad + Kk_{aux}(P + N) \\ &\approx k_{ct}(2K + 1)(P + N)[\log_2(P + N) + \mu'] \end{aligned} \quad (10)$$

where $\mu' = k_{aux}/2k_{ct}$. $Kk_{aux}(P + N)$ is the time required to complete auxiliary processes. These processes involve the multiplications of Eq. (3), the formation of the augmented sections $da_i(j)$, and the formation of $r(j)$ from the result sections $r_i(j)$. For the author's realization in which core memory was used for the secondary storage of input and output data, μ' was measured to be 1.5, which gives $k_{aux} = 3k_{ct} \approx 300\,\mu$sec. If slower forms of auxiliary storage were employed, this figure would be enlarged slightly.

For a specific pair of values M and N, P should be chosen to minimize T_{sect}. Since $P + N$ must be a power of 2, it is a simple matter to evaluate Eq. (10) for a few values of P that are compatible with this constraint and select the optimum choice. The size of available memory will place an additional constraint on how large $P + N$ may be allowed to become. Memory allocation considerations degrade the benefits of these methods when N becomes too large. In extreme cases one is forced to split the kernel, $s(j)$, into packets, each of which is considered separately. The results corresponding to all packets are then added together after each has been shifted by a suitable number of packet widths. For the author's realization N must be limited to occupy about ⅛ of the memory not used for the program or for the secondary storage of input/output data. For larger N, packets would be required.

COMBINATION OF SECTIONS IN PAIRS

If both functions to be convolved are real instead of complex, further time savings over Eq. (10) can be made by combining adjacent even and odd subscripted sections $da_i(j)$ into complex composites. Let even subscripted $da_i(j)$ be used as real parts and odd subscripted $da_{i+1}(j)$ be used as imaginary parts. Such a complex composite can then be transformed through the application of Eqs. (1a), (3), and (1b) to produce a complex composite result section. The desired even and odd subscripted result sections $r_i(j)$ and $r_{i+1}(j)$ are respectively the real and imaginary parts of that complex result section.

This device reduces the time required to perform the convolution by approximately a factor of 2. More precisely it modifies K by changing Eq. (8) to

$$K = \text{least integer} \geq M/2P \tag{11}$$

For very large numbers of sections, K, Eq. (10) can be simplified to a form involving M explicitly instead of implicitly through K. That form is given in Eq. (12)

$$T_{\text{fast}} \approx k_{ct} M((P + N)/P)[\log_2(P + N) + \mu'] \tag{12}$$

Since it makes no sense to choose $P < N$, for simple estimates of an approximate computation time we can write

$$T_{\text{fast}} \approx 2k_{ct}M[\log_2 N + \mu' + 1] \tag{13}$$

EMPIRICAL TIMES

The process for combined-sectioned-aperiodic convolution of real functions described above was implemented in the MAD language on the IBM 7094 Computer. Comparisons were made with a MAD language realization of a standard sum of lagged products for N = 16, 24, 32, 48, 64, 96, 128, 192, and 256. In each case M was selected to cause Eq. (11) to be fulfilled with the equal sign. This step favors the fast method by avoiding edge effects. However, P was not selected according to the optimization method described above (under "Sectioning Convolution"), but rather by selecting L as large as possible under the constraint.

$$\ln L \geq P/N \tag{14}$$

This choice can favor the standard method.

Table 1 compares for various N the actual computation times required in seconds as well as times in milliseconds per unit lag. Values of M, K, and L are also given.

Relative speed factors are shown in Table 2.

ACCURACY

The accuracy of the computational procedure described above is expected to be as good or better

Table 1. Comparative Convolution Times for Various N

N	16	24	32	48	64	96	128	192	256
M	192	208	384	416	768	832	1536	1664	3584
K	2	1	2	1	2	1	2	1	1
L	64	128	128	256	256	512	512	1024	2048
					Time in seconds				
T_{standard}	0.2	0.31	0.8	1.25	3.0	5.0	12	20	48
T_{fast}	0.3	0.4	0.6	0.8	1.3	1.8	3.0	3.8	8.0
					Time in milliseconds per unit lag				
$T_{\text{standard}/M}$	1.0	1.4	2.0	3.0	3.9	6.0	7.8	12.0	13.3
$T_{\text{fast}/M}$	1.5	1.9	1.5	1.9	1.6	2.1	1.9	2.2	2.2

Table 2. Speed Factors for Various N

N	16	24	32	48	64	96	128	192	256	512	1024	2048	4096
Speed factor	$\frac{2}{3}$	$\frac{3}{4}$	$\frac{4}{3}$	1.5	2.3	2.8	4.0	5.2	6	13*	24*	44*	80*

*Estimated values.

than that obtainable by summing products. Specific investigations of the accuracy of the program used to accumulate the data of Tables 1 and 2 are in process at the time of this writing. The above expectations are fostered by accuracy measurements made for floating-point data on the Cooley-Tukey procedure and a standard Fourier procedure. Since the standard Fourier procedure computes summed products, its accuracy characteristics are similar to those of a standard convolution which also computes summed products. Cases involving functions of period 64 and 256 were measured and it was discovered that two Cooley-Tukey transforms in cascade produced respectively as much, and half as much, error as a single standard Fourier transform. This data implies that the procedures disclosed here may yield more accurate results than standard methods with increasing relative accuracy for larger N.

APPLICATIONS

which, in turn, after weighting by so-called spectral windows are Fourier transformed into power spectrum estimates. Speed advantages can be gained when Eq. (15) is evaluated in a manner similar to that outlined above (under "Aperiodic Convolution") except that in this case L is only required to exceed $N + \Omega$ where Ω is the number of lags to be considered. This relaxed requirement on L is possible because it is not necessary to avoid the effect of performing the lags circularly for all L lags but rather for only Ω of them. An additional constraint is that Ω be larger than a multiple of $\log_2 L$. The usual practice is to evaluate Eq. (15) for a number of lags equal to a substantial fraction of N. Since the typical situation involves values of N in the hundreds and thousands, the associated savings may be appreciable for this application.

Digital spatial filtering is becoming an increasingly important subject.[3,4] The principles discussed here are easily extended to the computation of lagged products across two or more dimensions.

tions for the computa-
digital signal processing
filtering as it
ished through
on techniques.
by only a few
performed in
d for a stand-
escribed here.
d filter char-
realization by
on. The most
igh frequency
ation impulse
lse response is
nts. In these
he techniques
aded with dif-
ming spectral
f lagged prod-
) (15)

Time savings depend on the total number of data points contained within the entire data space in question, and they depend on this number in a manner similar to that characterizing the one-dimension case.

ACKNOWLEDGMENTS

The author is indebted to Charles M. Rader of the MIT Lincoln Laboratory for his ideas concerning the Cooley-Tukey algorithm and to Alan V. Oppenheim of the Electrical Engineering Department, MIT, for suggesting that high-speed convolutions might be realized through the utilization of that algorithm. During the preparation of this work the author became aware of the related independent efforts of Howard D. Helms, Bell Telephone Laboratories, and Gordon Sande, Jr., Princeton University.

REFERENCES

1. J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297-301, (Apr. 1965).

Today the major applica
tion of lagged products are
and spectral analysis.

Digital signal processing, or digital
is sometimes called, is often accompl
the use of suitable difference equatic
For difference equations characterized
parameters, computations may be
times short compared to those require
ard lagged product or the method d
However, in some cases, the desire
acteristics are too complex to permit
a sufficiently simple difference equatic
notable cases are those requiring h
selectivity coupled with short-dura
response and those in which the impu
found through physical measuremen
situations it is desirable to employ t
described here either alone or casca
ference equation filters.

The standard methods for perfor
analysis[2] involves the computation o
ucts of the form

$$F(j) = \sum_{J=0}^{N-j-1} x(J)y(J+j$$

2. R. B. Blackman and J. W. Tukey, *The Measurement of Power Spectra*, Dover Publications, New York, 1959; also *Bell System Technical Journal*, Jan. and Mar. 1958.

3. T. S. Huang and O. J. Tretiak, "Research in Picture Processing," *Optical and Electro-Optical Information Processing*, J. Tippett et al, eds., MIT Press, Cambridge, Mass., 1965, Chap. 3.

4. T. S. Huang, "PCM Picture Transmission," *IEEE Spectrum*, vol. 2, no. 12, pp. 57-63 (Dec. 1965).