The Chirp z-Transform Algorithm

L. R. RABINER, Member, IEEE
R. W. SCHAFER, Member, IEEE
Bell Telephone Laboratories, Inc.
Murray Hill, N. J.
C. M. RADER, Member, IEEE
Lincoln Laboratory[1]
Massachusetts Institute of Technology
Lexington, Mass.

Abstract

A computational algorithm for numerically evaluating the z-transform of a sequence of N samples is discussed. This algorithm has been named the chirp z-transform (CZT) algorithm. Using the CZT algorithm one can efficiently evaluate the z-transform at M points in the z-plane which lie on circular or spiral contours beginning at any arbitrary point in the z-plane. The angular spacing of the points is an arbitrary constant, and M and N are arbitrary integers.

The algorithm is based on the fact that the values of the z-transform on a circular or spiral contour can be expressed as a discrete convolution. Thus one can use well-known high-speed convolution techniques to evaluate the transform efficiently. For M and N moderately large, the computation time is roughly proportional to $(N+M) \log_2(N+M)$ as opposed to being proportional to $N \cdot M$ for direct evaluation of the z-transform at M points.

Manuscript received December 23, 1968; revised January 16, 1969.

This is a condensed version of a paper published in the *Bell System Technical Journal*, May 1969.

[1] Operated with support from the U. S. Air Force.

I. Introduction

In dealing with sampled data the z-transform plays the role which is played by the Laplace transform in continuous time systems. One example of its application is spectrum analysis. We shall see that the computation of sampled z-transforms, which has been greatly facilitated by the fast Fourier transform (FFT) [1], [2] algorithm, is still further facilitated by the chirp z-transform (CZT) algorithm to be described in this paper.

The z-transform of a sequence of numbers x_n is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x_n z^{-n}, \tag{1}$$

a function of the complex variable z. In general, both x_n and $X(z)$ could be complex. It is assumed that the sum on the right side of (1) converges for at least some values of z. We restrict ourselves to the z-transform of sequences with only a finite number N of nonzero points. In this case, we can rewrite (1) without loss of generality as

$$X(z) = \sum_{n=0}^{N-1} x_n z^{-n} \tag{2}$$

where the sum in (2) converges for all z except $z=0$.

Equations (1) and (2) are like the defining expressions for the Laplace transform of a train of equally spaced impulses of magnitudes x_n. Let the spacing of the impulses be T and let the train of impulses be $\sum_n x_n \delta(t-nT)$. Then the Laplace transform is $\sum_n x_n e^{-snT}$ which is the same as $X(z)$ if we let

$$z = e^{sT}. \tag{3}$$

If we are dealing with sampled waveforms the relation between the original waveform and the train of impulses is well understood in terms of the phenomenon of aliasing. Thus the z-transform of the sequence of samples of a time waveform is representative of the Laplace transform of the original waveform in a way which is well understood. The Laplace transform of a train of impulses repeats its values taken in a horizontal strip of the s-plane of width $2\pi/T$ in every other strip parallel to it. The z-transform maps each such strip into the entire z-plane, or conversely, the entire z-plane corresponds to any horizontal strip of the s-plane, e.g., the region $-\infty < \sigma < \infty$, $-\pi/T \leq \omega < \pi/T$, where $s = \sigma + j\omega$. In the same correspondence, the $j\omega$ axis of the s-plane, along which we generally equate the Laplace transform with the Fourier transform, is the unit circle in the z-plane, and the origin of the s-plane corresponds to $z=1$. The interior of the z-plane unit circle corresponds to the left half of the s-plane, and the exterior corresponds to the right half plane. Straight lines in the s-plane correspond to circles or spirals in the z-plane. Fig. 1 shows the correspondence of a contour in the s-plane to a contour in the z-plane. To evaluate the Laplace transform of the impulse train along the linear contour is to evaluate the z-transform of the sequence along the spiral contour.

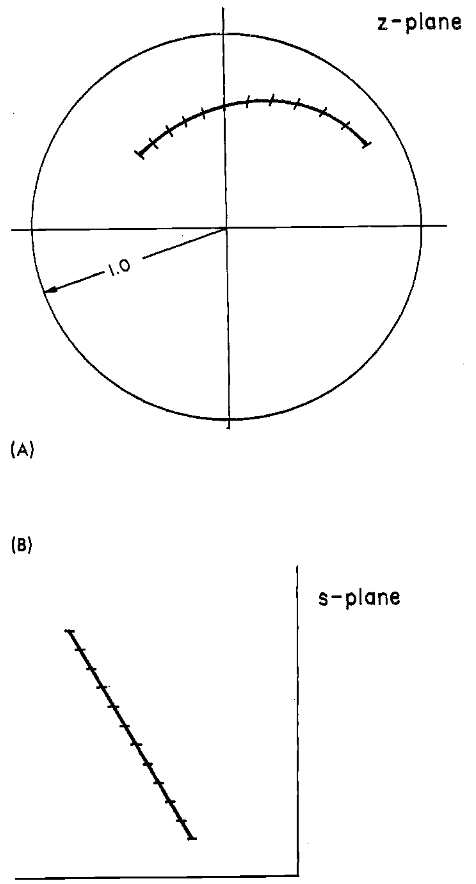


Fig. 1. The correspondence of (A) a z-plane contour to (B) an s-plane contour through the relation $z=e^{sT}$.

Values of the z-transform are usually computed along the path corresponding to the $j\omega$ axis, namely the unit circle. This gives the discrete equivalent of the Fourier transform and has many applications including the estimation of spectra, filtering, interpolation, and correlation. The applications of computing z-transforms off the unit circle are fewer, but one is presented elsewhere [6], namely the enhancement of spectral resonances in systems for which one has some foreknowledge of the locations of poles and zeroes.

Just as we can only compute (2) for a finite set of samples, so we can only compute (2) at a finite number of points, say z_k.

$$X_k = X(z_k) = \sum_{n=0}^{N-1} x_n z_k^{-n}. \tag{4}$$

The special case which has received the most attention is the set of points equally spaced around the unit circle,

$$z_k = \exp(j2\pi k/N), \qquad k = 0, 1, \cdots, N-1 \tag{5}$$

for which

$$X_k = \sum_{n=0}^{N-1} x_n \exp(-j2\pi k/N), \; k = 0, 1, \cdots, N-1. \tag{6}$$

Equation (6) is called the discrete Fourier transform (DFT). The reader may easily verify that, in (5), other values of k merely repeat the same N values of z_k, which are the Nth roots of unity. The discrete Fourier transform has assumed considerable importance, partly because of its nice properties, but mainly because since 1965 it has become widely known that the computation of (6) can be achieved, not in the N^2 complex multiplications and additions called for by direct application of (6), but in something of the order of $N \log_2 N$ operations if N is a power of two, or $N\sum_i m_i$ operations if the integers m_i are the prime factors of N. Any algorithm which accomplishes this is called an FFT. Much of the importance of the FFT is that DFT may be used as a stepping stone to computing lagged products such as convolutions, autocorrelations, and cross convolutions more rapidly than before [3], [4]. The DFT has, however, some limitations which can be eliminated using the CZT algorithm which we will describe. We shall investigate the computation of the z-transform on a more general contour, of the form

$$z_k = AW^{-k}, \qquad k = 0, 1, \cdots, M-1 \tag{7a}$$

where M is an arbitrary integer and both A and W are arbitrary complex numbers of the form

$$A = A_0 e^{j2\pi\theta_0} \tag{7b}$$

and

$$W = W_0 e^{j2\pi\phi_0}. \tag{7c}$$

(See Fig. 2.) The case $A=1$, $M=N$, and $W=\exp(-j2\pi/N)$ corresponds to the DFT. The general z-plane contour begins with the point $z=A$ and, depending on the value of W, spirals in or out with respect to the origin. If $W_0=1$, the contour is an arc of a circle. The angular spacing of the samples is $2\pi\phi_0$. The equivalent s-plane contour begins with the point

$$s_0 = \sigma_0 + j\omega_0 = \frac{1}{T} \ln A \tag{8}$$

and the general point on the s-plane contour is

$$s_k = s_0 + k(\Delta\sigma + j\Delta\omega) = \frac{1}{T}(\ln A - k \ln W), \quad k = 0, 1, \cdots, M-1. \tag{9}$$

Since A and W are arbitrary complex numbers we see that the points s_k lie on an arbitrary straight line segment of arbitrary length and sampling density. Clearly the contour indicated in (7a) is not the most general contour but it is considerably more general than that for which the DFT applies. In Fig. 2, an example of this more general contour is shown in both the z-plane and the s-plane.

To compute the z-transform along this more general contour would seem to require NM multiplications and additions as the special symmetries of $\exp(j2\pi k/N)$ which are exploited in the derivation of the FFT are absent in the more general case. However, we shall see that by using the sequence $W^{n^2/2}$ in various roles we can apply the FFT to the computation of the z-transform along the contour of (7a). Since for $W_0=1$, the sequence $W^{n^2/2}$ is a complex sinusoid of linearly increasing frequency, and since a

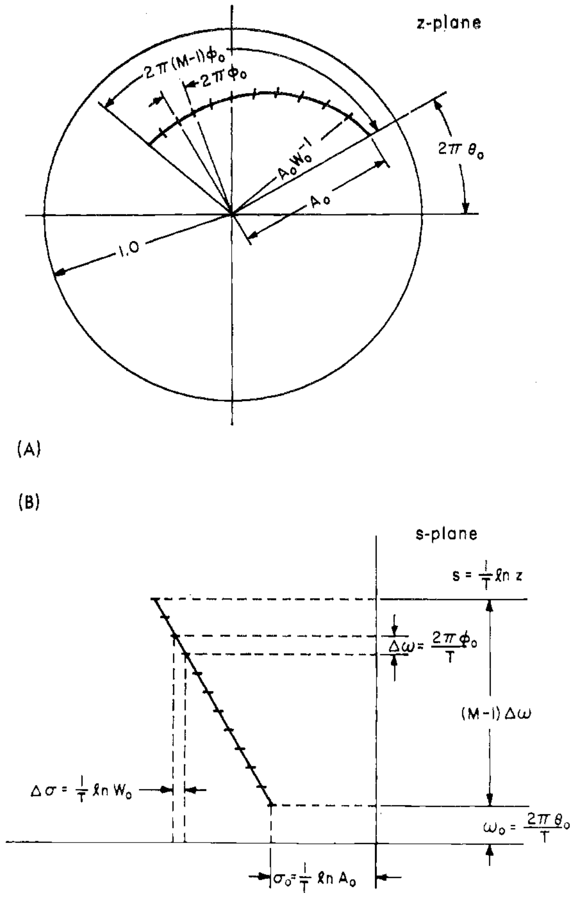


Fig. 2. An illustration of the independent parameters of the CZT algorithm. (A) How the z-transform is evaluated on a spiral contour starting at the point $z=A$. (B) The corresponding straight line contour and independent parameters in the s-plane.

similar waveform used in some radar systems has the picturesque name "chirp," we call the algorithm we are about to present the chirp z-transform (CZT). Since the CZT permits computing the z-transform on a more general tour than the FFT permits it is more flexible than the FFT, although it is also considerably slower. The additional freedoms offered by the CZT include the following:

1) The number of time samples does not have to equal the number of samples of the z-transform.

2) Neither M nor N need be a composite integer.

3) The angular spacing of the z_k is arbitrary.

4) The contour need not be a circle but can spiral in or out with respect to the origin. In addition, the point z_0 is arbitrary, but this is also the case with the FFT if the samples x_n are multiplied by z_0^{-n} before transforming.

II. Derivation of the CZT

Along the contour of (7a), (4) becomes

$$X_k = \sum_{n=0}^{N-1} x_n A^{-n} W^{nk}, \qquad k = 0, 1, \cdots, M-1 \quad (10)$$

which, at first appearance, seems to require NM complex multiplications and additions, as we have already ob-

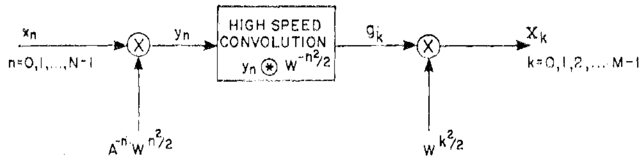


Fig. 3. An illustration of the steps involved in computing values of the z-transform using the CZT algorithm.

served. But, let us use the ingenious substitution, due to Bluestein [5],

$$nk = \frac{n^2 + k^2 - (k-n)^2}{2} \quad (11)$$

for the exponent of W in (10). This produces an apparently more unwieldy expression

$$X_k = \sum_{n=0}^{N-1} x_n A^{-n} W^{(n^2/2)} W^{(k^2/2)} W^{-(k-n)^2/2}, \quad k = 0, 1, \cdots, M-1; \quad (12)$$

but, in fact, (12) can be thought of as a three-step process consisting of:

1) forming a new sequence y_n by weighting the x_n according to the equation

$$y_n = x_n A^{-n} W^{n^2/2}, \quad n = 0, 1, \cdots, N-1; \quad (13)$$

2) convolving y_n with the sequence v_n defined as

$$v_n = W^{-n^2/2} \quad (14)$$

to give a sequence g_k,

$$g_k = \sum_{n=0}^{N-1} y_n v_{k-n}, \quad k = 0, 1, \cdots, M-1; \quad (15)$$

3) multiplying g_k by $W^{k^2/2}$ to give X_k,

$$X_k = g_k W^{k^2/2}, \quad k = 0, 1, \cdots, M-1. \quad (16)$$

The three-step process is illustrated in Fig. 3. Steps 1 and 3 require N and M multiplications, respectively, and step 2 is a convolution which may be computed by the high-speed technique disclosed by Stockham [3], based on the use of the FFT. Step 2 is the major part of the computational effort and requires a time roughly proportional to $(N+M) \log (N+M)$.

Bluestein employed the substitution of (11) to convert a DFT to a convolution as in Fig. 3. The linear system to which the convolution is equivalent can be called a chirp filter which is, in fact, also sometimes used to resolve a spectrum. Bluestein [5] showed that for N a perfect square, the chirp filter could be synthesized recursively with $\sqrt{N}$ multipliers and the computation of a DFT could then be proportional to $N^{3/2}$.

The flexibility and speed of the CZT algorithm are related to the flexibility and speed of the method of high-speed convolution using the FFT. The reader should recall that the product of the DFT's of two sequences is the DFT of the circular convolution of the two sequences and, therefore, a circular convolution is computable as two DFT's, the multiplication of two arrays of complex numbers, and an inverse discrete Fourier transform (IDFT), which can also be computed by the FFT. Ordinary convolutions can be computed as circular convolutions by appending zeroes to the end of one or both sequences so that the correct numerical answers for the ordinary convolution can result from a circular convolution.

We shall now summarize the details of the CZT algorithm on the assumption that an already existing FFT program (or special-purpose machine) is available to compute DFT's and IDFT's.

Begin with a waveform in the form of N samples x_n and seek M samples of X_k where A and W have also been chosen.

1) Choose L, the smallest integer greater than or equal to $N+M-1$ which is also compatible with our high-speed FFT program. For most users this will mean L is a power of two. Note that while many FFT programs will work for arbitrary L, they are not equally efficient for all L. At the very least, L should be highly composite.

2) Form an L point sequence y_n from x_n by

$$y_n = \begin{cases} A^{-n}W^{n^2/2}x_n & n=0, 1, 2, \cdots, N-1 \\ 0 & n=N, N+1, \cdots, L-1. \end{cases} \tag{17}$$

3) Compute the L point DFT of y_n by the FFT. Call this Y_r, $r=0, 1, \cdots, L-1$.

4) Define an L point sequence v_n by the relation

$$v_n = \begin{cases} W^{-n^2/2} & 0 \leq n \leq M-1 \\ W^{-(L-n)^2/2} & L-N+1 \leq n < L \\ \text{arbitrary} & \text{other } n, \text{ if any.} \end{cases} \tag{18}$$

Of course, if L is exactly equal to $M+N-1$, the region in which v_n is arbitrary will not exist. If the region does exist an obvious possibility is to increase M, the desired number of points of the z-transform we compute, until the region does not exist.

Note that v_n could be cut into two with a cut between $n=M-1$ and $n=L-N+1$ and if the two pieces were abutted together differently, the resulting sequence would be a slice out of the indefinite length sequence $W^{-n^2/2}$. This is illustrated in Fig. 4. The sequence v_n is defined the way it is in order to force the circular convolution to give us the desired numerical results of an ordinary convolution.

5) Compute the DFT of v_n and call it V_r, $r=0, 1, \cdots, L-1$.

6) Multiply V_r and Y_r point by point, giving G_r:

$$G_r = V_r Y_r, \quad r = 0, 1, \cdots, L-1.$$

7) Compute the L point IDFT g_k, of G_r.

8) Multiply g_k by $W^{k^2/2}$ to give the desired X_k:

$$X_k = W^{k^2/2} g_k, \quad k = 0, 1, 2, \cdots, M-1.$$

The g_k for $k \geq M$ are discarded.

Fig. 4 represents typical waveforms (magnitudes shown, phase omitted) involved in each step of the process.

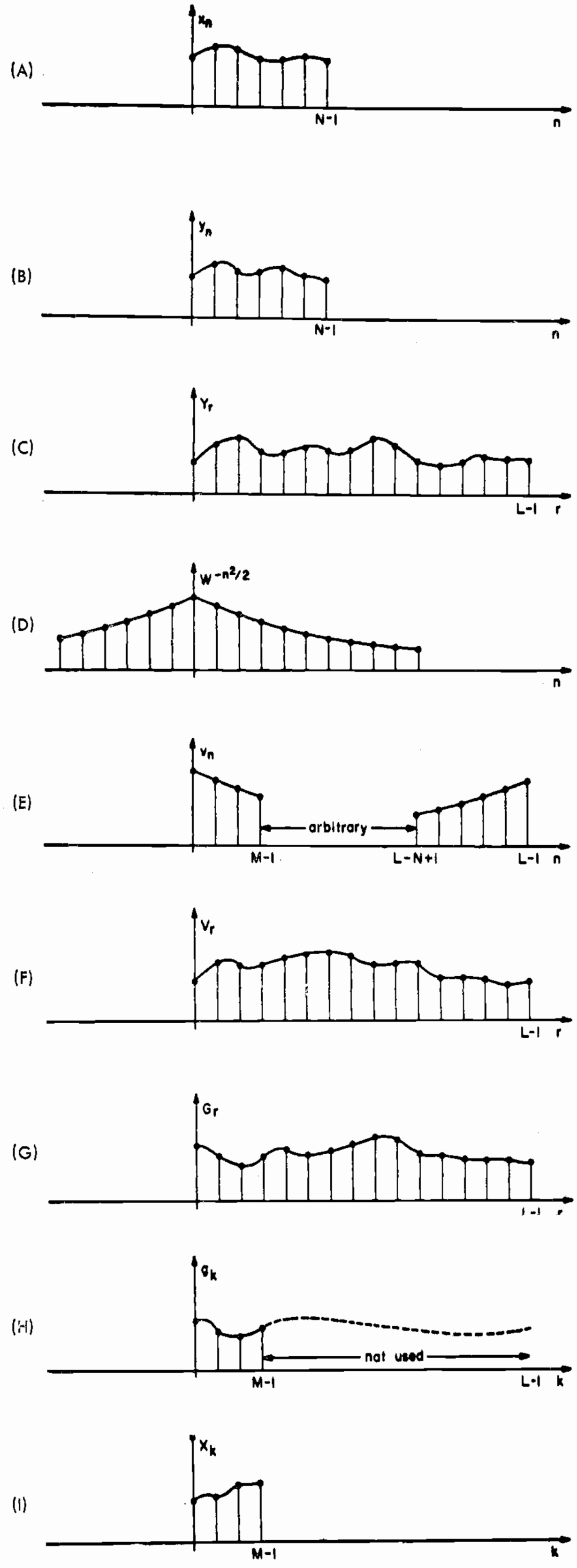


Fig. 4. Schematic representation of the various sequences involved in the CZT algorithm. (A) The input sequence x_n with N values. (B) The weighted input sequence $y_n = A^{-n}W^{n^2/2}x_n$. (C) The DFT of y_n. (D) The values of the indefinite sequence $W^{-n^2/2}$. (E) The sequence v_n formed appropriately from segments of $W^{-n^2/2}$. (F) The DFT of v_n. (G) The product $G_r = Y_r \cdot V_r$. (H) The IDFT of G_r. (I) The desired M values of the z-transform.

III. Fine Points of the Computation

Operation Count and Timing Considerations

An operation count can be made, roughly, from the eight steps just presented. We will give it step by step because there are, of course, many possible variations to be considered.

1) We assume that step 1, choosing L, is a negligible operation.

2) Forming y_n from x_n requires N complex multiplications, not counting the generation of the constants $A^{-n}W^{n^2/2}$. The constants may be prestored, computed as needed, or generated recursively as needed. The recursive computation would require two complex multiplications per point.

3) An L point DFT requires a time $k_{\text{FFT}}L \log_2 L$ for L a power of two, and a very simple FFT program. More complicated (but faster) programs have more complicated computing time formulas.

4), 5) v_n is computed for either M or N points, whichever is greater. The symmetry in $W^{-n^2/2}$ permits the other values of v_n to be obtained without computation. Again, v_n can be computed recursively. The FFT takes the same time as that in step 3. If the same contour is used for many sets of data, V_r need only be computed once, and stored.

6) This step requires L complex multiplications.

7) This is another FFT and requires the same time as step 3.

8) This step requires M complex multiplications.

As the number of samples of x_n or X_k grow large, the computation time for the CZT grows asymptotically as something proportional to $L \log_2 L$. This is the same sort of asymptotic dependence of the FFT, but the constant of proportionality is bigger for the CZT because two or three FFT's are required instead of one, and because L is greater than N or M. Still, the CZT is faster than the direct computation of (10) even for relatively modest values of M and N, of the order of 50.

Reduction in Storage

The CZT can be put into a more useful form for computation by redefining the substitution of (11) to read

$$nk = \frac{(n - N_0)^2 + k^2 - (k - n + N_0)^2 + 2N_0 k}{2}.$$

Equation (12) can now be rewritten as

$$X_k = W^{k^2/2}W^{N_0 k} \sum_{n=0}^{N-1} x_n A^{-n} W^{(n-N_0)^2/2} W^{-(k-n+N_0)^2/2}.$$

The form of the new equation is similar to (12) in that the input data x_n are pre-weighted by a complex sequence $(A^{-n}W^{(n-N_0)^2/2})$, convolved with a second sequence $(W^{-(n-N_0)^2/2})$, and post-weighted by a third sequence $(W^{k^2/2}W^{N_0 k})$ to compute the output sequence X_k. However, there are differences in the detailed procedures for realizing the CZT. The input data x_n can be thought of as having been shifted by N_0 samples to the left; e.g., x_0 is weighted by $W^{N_0^2/2}$ instead of W_0. The region over which $W^{-n^2/2}$ must be formed, in order to obtain correct results from the convolution, is

$$-N + 1 + N_0 \leq n \leq M - 1 + N_0.$$

By choosing $N_0 = (N-M)/2$ it can be seen that the limits over which $W^{-n^2/2}$ is evaluated are symmetric; i.e., $W^{-n^2/2}$ is a symmetric function in both its real and imaginary parts. (It follows thus that the transform of $W^{-n^2/2}$ is also symmetric in both its real and imaginary parts.) It can be shown that using this special value of N_0, only $(L/2+1)$ points of $W^{-n^2/2}$ need be calculated and stored and these $(L/2+1)$ complex points can be transformed using an $L/2$ point transform.[2] Hence the total storage required for the transform of $W^{-n^2/2}$ is $L+2$ locations.

The only other modifications to the detailed procedures for evaluating the CZT presented in Section II of this paper are: 1) following the L point IDFT of step 7, the data of array g_k must be rotated to the left by N_0 locations; and 2) the weighting factor of the g_k is $W^{k^2/2}W^{N_0 k}$ rather than $W^{k^2/2}$. The additional factor $W^{N_0 k}$ represents a data shift of N_0 samples to the right, thus compensating the initial shift and keeping the effective positions of the data invariant to the value of N_0 used.

An estimate of the storage required to perform the CZT can now be made. Assuming that the entire process is to take place in core, storage is required for V_r which takes $L+2$ locations; for y_n, which takes $2L$ locations; and perhaps for some other quantities which we wish to save, e.g., the input, or values of $W^{+n^2/2}$ or $A^{-n}W^{n^2/2}$.

Additional Considerations

Since the CZT permits $M \neq N$, it is possible that occasions will arise where $M \gg N$ or $N \gg M$. In these cases, if the smaller number is small enough, the direct method of (10) is called for. However, if even the smaller number is large it may be appropriate to use the methods of sectioning described by Stockham [3]. Either the lap-save or lap-add methods may be used. Sectioning may also be used when problems too big to be handled in core memory arise. We have not actually encountered any of these problems and have not programmed the CZT with provision for sectioning.

Since the contour for the CZT is a straight line segment in the s-plane, it is apparent that repeated application of the CZT can compute the z-transform along a contour which is piecewise spiral in the z-plane or piecewise linear in the s-plane.

Let us briefly consider the CZT algorithm for the case of z_k all on the unit circle. This means that the z-transform is like a Fourier transform. Unlike the DFT, which by definition gives N points of transform for N points of

[2] The technique for transforming two real symmetric L point sequences using one $L/2$ point FFT was demonstrated by J. Cooley at the FFT Workshop, Arden House, October 1968. A summary of this technique is presented in the Appendix.

data, the CZT does not require $M=N$. Furthermore, the z_k need not stretch over the entire unit circle but can be equally spaced along an arc. Let us assume, however, that we are really interested in computing the N point DFT of N data points. Still the CZT permits us to choose any value of N, highly composite, somewhat composite, or even prime, without strongly affecting the computation time. An important application of the CZT may be computing DFT's when N is not a power of two and when the program or special-purpose device available for computing DFT's by FFT is limited to the case of N a power of two.

There is also no reason why the CZT cannot be extended to the case of transforms in two or more dimensions with similar considerations. The two-dimensional DFT becomes a two-dimensional convolution which is computable by FFT techniques.

We caution the reader to note that for the ordinary FFT the starting point of the contour is still arbitrary; merely multiply the waveform x_n by A^{-n} before using the FFT, and the first point on the contour is effectively moved from $z=1$ to $z=A$. However, the contour is still restricted to a circle concentric with the origin. The angular spacing of z_k for the FFT can also be controlled to some extent by appending zeroes to the end of x_n before computing the DFT (to decrease the angular spacing of the z_k) or by choosing only P of the N points x_n and adding together all the x_n for which the n are congruent modulo P; i.e., wrapping the waveform around a cylinder and adding together the pieces which overlap (to increase the angular spacing).

IV. Limitations

One limitation in using the CZT algorithm to evaluate the z-transform off the unit circle stems from the fact that we may be required to compute $W_0^{\pm n^2/2}$ for large n. If W_0 differs very much from 1.0, $W_0^{\pm n^2/2}$ can become very large or very small when n becomes large. (We require a large n when either M or N become large, since we need to evaluate $W^{n^2/2}$ for n in the range $-N<n<M$.) For example, if $W_0=e^{-.25/1000}\approx 0.999749$, and $n=1000$, $W_0^{\pm n^2/2}=e^{\pm 125}$ which exceeds the single precision floating point capability of most computers by a large amount. Hence the tails of the functions $W^{\pm n^2/2}$ can be greatly in error, thus causing the tails of the convolution (the high frequency terms) to be grossly inaccurate. The low frequency terms of the convolution will also be slightly in error but these errors are negligible in general.

The limitation on contour distance in or out from the unit circle is again due to computation of $W^{\pm n^2/2}$. As W_0 deviates significantly from 1.0, the number of points for which $W^{\pm n^2/2}$ can be accurately computed decreases. It is of importance to stress, however, that for $W_0=1$, there is no limitation of this type since $W^{\pm n^2/2}$ is always of magnitude 1.

The other main limitation on the CZT algorithm stems from the fact that two L point, and one $L/2$ point, FFT's must be evaluated where L is the smallest convenient integer greater than $N+M-1$ as mentioned previously. We need one FFT and $2L$ storage locations for the transform of $x_nA^{-n}W^{n^2/2}$; one FFT and $L+2$ storage locations for the transform $W^{-n^2/2}$; and one FFT for the inverse transform of the product of these two transforms. We do not know a way of computing the transform of $W^{-n^2/2}$ either recursively or by a specific formula (except in some trivial cases). Thus we must compute this transform and store it in an extra $L+2$ storage locations. Of course, if many transforms are to be done with the same value of L, we need not compute the transform of $W^{-n^2/2}$ each time.

We can compute the quantities $A^{-n}W^{n^2/2}$ recursively as they are needed to save computation and storage. This is easily seen from the fact that

$$A^{-(n+1)}W^{(n+1)^2/2} = (A^{-n}W^{n^2/2})\cdot W^nW^{1/2}A^{-1}. \quad (19)$$

If we define

$$C_n = A^{-n}W^{n^2/2} \quad (20)$$

and

$$D_n = W^nW^{1/2}A^{-1} \quad (21)$$

then

$$D_{n+1} = W\cdot D_n \quad (22)$$

and

$$C_{n+1} = C_n\cdot D_n. \quad (23)$$

Setting $A=1$ in (19) to (23) provides an algorithm for the coefficients required for the output sequence. A similar recursion formula can be obtained for generating the sequence $A^{-n}W^{(n-N_0)^2/2}$. The user is cautioned that recursive computation of these coefficients may be a major source of numerical error, especially when $W_0\approx 1$, or $\phi_0\approx 0$.

V. Summary

A computational algorithm for numerically evaluating the z-transform of a sequence of N time samples was presented. This algorithm, entitled the chirp z-transform algorithm, enables the evaluation of the z-transform at M equi-angularly spaced points on contours which spiral in or out (circles being a special case) from an arbitrary starting point in the z-plane. In the s-plane the equivalent contour is an arbitrary straight line.

The CZT algorithm has great flexibility in that neither N or M need be composite numbers; the output point spacing is arbitrary; the contour is fairly general and N need not be the same as M. The flexibility of the CZT algorithm is due to being able to express the z-transform on the above contours as a convolution, permitting the use of well-known high-speed convolution techniques to evaluate the convolution.

Applications of the CZT algorithm include enhancement of poles for use in spectral analysis; high resolution, narrowband frequency analysis; and time interpolation of data from one sampling rate to any other sampling rate. These applications are explained in detail elsewhere [6]. The CZT algorithm also permits use of a radix-2 FFT program or device to compute the DFT of an arbitrary number of samples. Examples illustrating how the CZT algorithm is used in specific cases are included elsewhere [6]. It is anticipated that other applications of the CZT algorithm will be found.

Appendix

The purpose of this Appendix is to show how the FFT's of two real, symmetric L point sequences can be obtained using one $L/2$ point FFT.

Let x_n and y_n be two real, symmetric L point sequences with corresponding DFT's X_k and Y_k. By definition,

$$\begin{aligned} x_n &= x_{L-n} \\ y_n &= y_{L-n} \end{aligned} \qquad n = 0, 1, 2, \cdots, L-1,$$

and it is easily shown that X_k and Y_k are real, symmetric L point sequences, so that

$$\begin{aligned} X_k &= X_{L-k} \\ Y_k &= Y_{L-k} \end{aligned} \qquad k = 0, 1, 2, \cdots, L-1.$$

Define a complex $L/2$ point sequence u_n whose real and imaginary parts are

$$\left.\begin{aligned} \text{Re}\,[u_n] &= x_{2n} - y_{2n+1} + y_{2n-1} \\ \text{Im}\,[u_n] &= y_{2n} + x_{2n+1} - x_{2n-1} \end{aligned}\right\} \quad n = 0, 1, \cdots, L/2-1.$$

The $L/2$ point DFT of u_n is denoted U_k and is calculated by the FFT. The values of X_k and Y_k may be computed from U_k using the relations

$$X_k = \tfrac{1}{2}\{\text{Re}\,[U_k] + \text{Re}\,[U_{L/2-k}]\} - \frac{1}{4 \sin \frac{2\pi}{L} k} \{\text{Re}\,[U_k] - \text{Re}\,[U_{L/2-k}]\}$$

$$Y_k = \tfrac{1}{2}\{\text{Im}\,[U_k] + \text{Im}\,[U_{L/2-k}]\} - \frac{1}{4 \sin \frac{2\pi}{L} k} \{\text{Im}\,[U_k] - \text{Im}\,[U_{L/2-k}]\}$$

for $k = 1, 2, \cdots, L/2-1$. The remaining values of X_k and Y_k are obtained from the relations

$$X_0 = \sum_{n=0}^{L-1} x_n$$

$$Y_0 = \sum_{n=0}^{L-1} y_n$$

$$X_{L/2} = \sum_{n=0}^{L-1} x_n(-1)^n$$

$$Y_{L/2} = \sum_{n=0}^{L-1} y_n(-1)^n.$$

References

[1] J. W. Cooley and J. W. Tukey, "An algorithm foı the machine calculation of complex Fourier series," *Math. Comp.*, vol. 19, pp. 297–301, 1965.

[2] G-AE Subcommittee on Measurement Concepts, "What is the fast Fourier transform?" *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, pp. 45–55, June 1967.

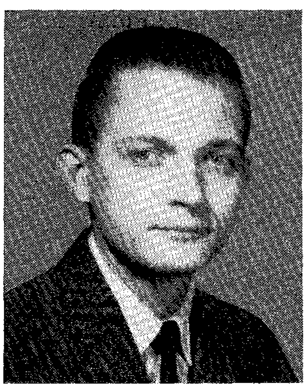
[3] T. G. Stockham, Jr., "High speed convolution and correlation," *1966 Spring Joint Computer Conf., AFIPS Proc.*, vol. 28. Washington, D. C.: Spartan, 1966, pp. 229–233.

[4] H. D. Helms, "Fast Fourier transform method of computing difference equations and simulating filters," *IEEE Trans. Audio and Electroacoustics*, vol. AU-15, pp. 85–90, June 1967.

[5] L. I. Bluestein, "A linear filtering approach to the computation of the discrete Fourier transform," *1968 NEREM Rec.*, pp. 218–219.

[6] L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp *z*-transform algorithm and its applications," *Bell Sys. Tech. J.*, vol. 48, pp. 1249–1292, May 1969.

Lawrence R. Rabiner (S'62–M'67), for a photograph and biography, please see page 13 of the March, 1969, issue of this TRANSACTIONS.



Ronald W. Schafer (S'62–M'67) was born in Tecumseh, Neb., on February 17, 1938. He received the B.Sc.E.E. and M.Sc.E.E. degrees from the University of Nebraska, Lincoln, in 1961 and 1962, respectively, and the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in February, 1968.

From 1962 to 1963 he served as an Instructor in the Department of Electrical Engineering at the University of Nebraska, and from 1964 until 1968 he was an Instructor in the Department of Electrical Engineering at M.I.T., where he received a departmental award for teaching. During 1965 he was with the M.I.T. Electronic Systems Laboratory, where he worked on digital computer simulations of guidance and control systems, and from 1966 to 1968 he was with the M.I.T. Research Laboratory of Electronics. At present he is engaged in research on digital speech processing techniques at Bell Telephone Laboratories, Inc., Murray Hill, N. J.

Dr. Schafer is a member of Eta Kappa Nu, Sigma Xi, and the Acoustical Society of America.

C. M. Rader (S'59-M'62), for a photograph and biography, please see this issue, page 76.