Chirp Z-transform

From Wikipedia, the free encyclopedia
(Redirected from Bluestein's FFT algorithm)

The **Chirp Z-transform** (**CZT**) is a generalization of the discrete Fourier transform. While the DFT samples the Z plane at uniformly-spaced points along the unit circle, the chirp Z-transform samples along spiral arcs in the Z-plane, corresponding to straight lines in the S plane.[1][2] The DFT, real DFT, and zoom DFT can be calculated as special cases of the CZT.

Specifically, the chirp Z transform calculates the Z transform at a finite number of points z_k along a logarithmic spiral contour, defined as:[3][1]

$$z_k = A \cdot W^{-k}, k = 0, 1, \dots, M-1$$

where A is the complex starting point, W is the complex ratio between points, and M is the number of points to calculate.

Contents [hide]

Bluestein's algorithm [edit]

Bluestein's algorithm[4] expresses the CZT as a convolution and implements it efficiently using FFT/IFFT.

As the DFT is a special case of the CZT, this allows the efficient calculation of discrete Fourier transform (DFT) of arbitrary sizes, including prime sizes. (The other algorithm for FFTs of prime sizes, Rader's algorithm, also works by rewriting the DFT as a convolution.) It was conceived in 1968 by Leo Bluestein.[5] Bluestein's algorithm can be used to compute more general transforms than the DFT, based on the (unilateral) z-transform (Rabiner *et al.*, 1969).

Recall that the DFT is defined by the formula

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk} \qquad k = 0, \dots, N-1.$$

If we replace the product nk in the exponent by the identity

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

we thus obtain:

$$X_k = e^{-\frac{\pi i}{N} k^2} \sum_{n=0}^{N-1} \left(x_n e^{-\frac{\pi i}{N} n^2} \right) e^{\frac{\pi i}{N} (k-n)^2} \qquad k = 0, \dots, N-1.$$

This summation is precisely a convolution of the two sequences a_n and b_n defined by:

$$a_n = x_n e^{-\frac{\pi i}{N} n^2}$$
$$b_n = e^{\frac{\pi i}{N} n^2},$$

with the output of the convolution multiplied by N phase factors b_k^*. That is:

$$X_k = b_k^* \sum_{n=0}^{N-1} a_n b_{k-n} \qquad k = 0, \dots, N-1.$$

This convolution, in turn, can be performed with a pair of FFTs (plus the pre-computed FFT of complex chirp b_n) via the convolution theorem. The key point is that these FFTs are not of the same length N: such a convolution can be computed exactly from FFTs only by zero-padding it to a length greater than or equal to $2N-1$. In particular, one can pad to a power of two or some other highly composite size, for which the FFT can be efficiently performed by e.g. the Cooley–Tukey algorithm in $O(N \log N)$ time. Thus, Bluestein's algorithm provides an $O(N \log N)$ way to compute prime-size DFTs, albeit several times slower than the Cooley–Tukey algorithm for composite sizes.

The use of zero-padding for the convolution in Bluestein's algorithm deserves some additional comment. Suppose we zero-pad to a length $M \geq 2N-1$. This means that a_n is extended to an array A_n of length M, where $A_n = a_n$ for $0 \leq n < N$ and $A_n = 0$ otherwise—the usual meaning of "zero-padding". However, because of the b_{k-n} term in the convolution, both positive and *negative* values of n are required for b_n (noting that $b_{-n} = b_n$). The periodic boundaries implied by the DFT of the zero-padded array mean that $-n$ is equivalent to $M-n$. Thus, b_n is extended to an array B_n of length M, where $B_0 = b_0$, $B_n = B_{M-n} = b_n$ for $0 < n < N$, and $B_n = 0$ otherwise. A and B are then FFTed, multiplied pointwise, and inverse FFTed to obtain the convolution of a and b, according to the usual convolution theorem.

Let us also be more precise about what type of convolution is required in Bluestein's algorithm for the DFT. If the sequence b_n were periodic in n with period N, then it would be a cyclic convolution of length N, and the zero-padding would be for computational convenience only. However, this is not generally the case:

$$b_{n+N} = e^{\frac{\pi i}{N}(n+N)^2} = b_n e^{\frac{\pi i}{N}(2Nn+N^2)} = (-1)^N b_n.$$

Therefore, for N even the convolution is cyclic, but in this case N is composite and one would normally use a more efficient FFT algorithm such as Cooley–Tukey. For N odd, however, then b_n is antiperiodic and we technically have a negacyclic convolution of length N. Such distinctions disappear when one zero-pads a_n to a length of at least $2N-1$ as described above, however. It is perhaps easiest, therefore, to think of it as a subset of the outputs of a simple linear convolution (i.e. no conceptual "extensions" of the data, periodic or otherwise).

z-Transforms [edit]

Bluestein's algorithm can also be used to compute a more general transform based on the (unilateral) z-transform (Rabiner *et al.*, 1969). In particular, it can compute any transform of the form:

$$X_k = \sum_{n=0}^{N-1} x_n z^{nk} \qquad k = 0, \dots, M-1,$$

for an *arbitrary* complex number z and for *differing* numbers N and M of inputs and outputs. Given Bluestein's algorithm, such a transform can be used, for example, to obtain a more finely spaced interpolation of some portion of the spectrum (although the frequency resolution is still limited by the total sampling time, similar to a Zoom FFT), enhance arbitrary poles in transfer-function analyses, etc.

The algorithm was dubbed the *chirp* z-transform algorithm because, for the Fourier-transform case ($|z| = 1$), the sequence b_n from above is a complex sinusoid of linearly increasing frequency, which is called a (linear) chirp in radar systems.

References [edit]

1. ^ [a] [b] A study of the Chirp Z-transform and its applications - Shilling, Steve Alan
2. ^ http://www.mathworks.com/help/signal/ref/czt.html
3. ^ http://prod.sandia.gov/techlib/access-control.cgi/2005/057084.pdf
4. ^ "Bluestein's FFT Algorithm". DSPRelated.com.
5. ^ "A linear filtering approach to the computation of discrete Fourier transform". *Audio and Electroacoustics* (IEEE Transactions) **4**: 451–455. 1970.

- Leo I. Bluestein, "A linear filtering approach to the computation of the discrete Fourier transform," *Northeast Electronics Research and Engineering Meeting Record* **10**, 218-219 (1968).
- Lawrence R. Rabiner, Ronald W. Schafer, and Charles M. Rader, "The chirp z-transform algorithm and its application," *Bell Syst. Tech. J.* **48**, 1249-1292 (1969). Also published in: Rabiner, Shafer, and Rader, "The chirp z-transform algorithm," *IEEE Trans. Audio Electroacoustics* **17** (2), 86–92 (1969).
- D. H. Bailey and P. N. Swarztrauber, "The fractional Fourier transform and applications," *SIAM Review* **33**, 389-404 (1991). (Note that this terminology for the z-transform is nonstandard: a fractional Fourier transform conventionally refers to an entirely different, continuous transform.)
- Lawrence Rabiner, "The chirp z-transform algorithm—a lesson in serendipity," *IEEE Signal Processing Magazine* **21**, 118-119 (March 2004). (Historical commentary.)

External links [edit]

- A DSP algorithm for frequency analysis - the Chirp-Z Transform (CZT)

Categories: FFT algorithms