OPEN

Numerical error analysis of the ICZT algorithm for chirp contours on the unit circle

Vladimir Sukhoy & Alexander Stoytchev *

This paper shows that the inverse chirp z-transform (ICZT), which generalizes the inverse fast Fourier transform (IFFT) off the unit circle in the complex plane, can also be used with chirp contours that perform partial or multiple revolutions on the unit circle. This is done as a special case of the ICZT, which in algorithmic form has the same computational complexity as the IFFT, i.e., $O(n \log n)$. Here we evaluate the ICZT algorithm for chirp contours on the unit circle and show that it is numerically accurate for large areas of the parameter space. The numerical error in this case depends on the polar angle between two adjacent contour points. More specifically, the error profile for a transform of size n is determined by the elements of the Farey sequence of order $n - 1$. Furthermore, this generalization allows the use of non-orthogonal frequency components, thus lifting one of the main restrictions of the IFFT.

The *Inverse Chirp Z-Transform* (ICZT) is a generalization of the *Inverse Fast Fourier Transform* (IFFT), which is one of the most popular and useful algorithms^{1,2}. The sampling points used by the ICZT are distributed along a logarithmic spiral contour in the complex plane. The shape of this contour is determined by the complex parameters A and W such that the k -th contour point is equal to AW^{-k} , where k is a zero-based index (see Supplementary Section S1).

This paper studies the properties of the ICZT for the special case when the magnitudes of A and W are equal to 1, which restricts the contour to lie on the unit circle. Unlike the IFFT, the ICZT can work with a contour that performs a partial revolution, a full revolution, or more than one revolution. The effect of this generalization is that the frequency components specified by the sampling points are no longer restricted to be harmonically related or orthogonal. Lifting this restriction makes it possible to use the spectrum more efficiently.

Many applications require both *signal analysis* and *signal synthesis*. Traditionally, these tasks have been performed with the FFT and IFFT algorithms that were published in 1965³. Both algorithms run in $O(n \log n)$ time, which makes them fast and practical. The *Chirp Z-Transform* (CZT), which generalizes the *Fast Fourier Transform* (FFT) and also runs in $O(n \log n)$ time, was discovered in 1969^{4–9}. The inverse algorithm, however, remained elusive for the next 50 years. The ICZT algorithm also runs in $O(n \log n)$ time¹⁰, where n is the size of the transform. This enables applications in which the CZT is paired with the ICZT similarly to how the FFT is often paired with the IFFT. Application domains that could benefit from this include signal processing, electronics, medical imaging, radar, sonar, wireless communications, and others.

Figure 1 shows three examples of 16-point chirp contours that lie on the unit circle. The spacing between the 16 points is different in each plot. From left to right, the angular interval between neighboring points is equal to: 5.625°, 11.25°, and 22.5°. The last contour corresponds to the FFT after reordering the output vector elements (see Supplementary Sections S1 and S2).

Figure 2 shows the numerical error for the sequential application of the CZT followed by the ICZT for a transform of size 16. Each point represents the average numerical error for 10 randomly generated unit-length complex input vectors. The numerical error is plotted as a function of the polar angle of the transform parameter W . The red points indicate the numerical error for the three contours from Fig. 1. For these contours, the error decreases as the samples cover larger fractions of the circle. The error for the rightmost contour is very small and is close to the machine epsilon. Increasing the angle of W above 22.5° eventually wraps the chirp contour over the unit circle more than once (see Fig. 3). The behavior of the error in those cases is more complicated and is related to Farey sequences.

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, 50011, USA. *email: alexs@iastate.edu

Figure 1. Three chirp contours on the unit circle, each with 16 points. The angular interval between any two adjacent points is: $90^\circ/16$, $180^\circ/16$, and $360^\circ/16$. The starting point of each contour is indicated with an unfilled circle.

Figure 2. Absolute numerical error for 16-point chirp contours, shown as a function of the polar angle of W , discretized at $360^\circ/2048$. Each point represents the average error for ten random input vectors, each of unit length. The three red points indicate the errors for the three chirp contours shown in Fig. 1.

Related Work

There have been several unsuccessful attempts^{11–14} to derive an efficient inverse chirp z-transform (ICZT) algorithm. Most of these attempts have focused on the special case of inverting the CZT for chirp contours on the unit circle. In one case¹³, a modified version of the forward transform, in which the circular chirp contour was traversed in the opposite direction, was presented as the ICZT transform. This approach, however, does not really invert the forward CZT in the general case, i.e., for logarithmic spiral contours. This approach also doesn't work for all chirp contours that lie on the unit circle, because the inverse of the Vandermonde matrix used by the forward transform is not, in general, a Vandermonde matrix. Traversing the contour in reverse order is equivalent to using another Vandermonde matrix as the transformation matrix. Our paper provides experimental results that show that this method does not work.

The literature describes two algorithms that generalize the FFT on the unit circle: the *Chirp Transform Algorithm* (CTA)¹⁵ and the *Fractional Fourier Transform* (FRFT)^{16,17}. Both algorithms are special cases of the CZT for chirp contours that lie on the unit circle, but the FRFT contours always start at the complex point $(1, 0)$. It has been estimated¹⁸ that the inverse chirp transform can be performed in $2M(n) + O(n)$, where $M(n)$ can be taken in $O(n \log(n) \log(\log(n)))$.

As described in Supplementary Section S3, the CTA and the FRFT can each be implemented with a single call to the CZT algorithm. Supplementary Section S3 also describes how to implement the inverse CTA and the inverse FRFT algorithms, which have not been described in the literature until now. We named these algorithms ICTA and IFRFT. They are implemented as special cases of the ICZT algorithm. Both algorithms run in $O(n \log n)$ time and use $O(n)$ memory.

The FFT and IFFT are two very similar algorithms. They are also very stable numerically. The reason for this is that the harmonically-spaced frequency components that they use are orthogonal. This condition doesn't hold for the CZT and the ICZT, which explains why the ICZT algorithm is substantially different from the CZT algorithm. The numerical accuracy of the ICZT depends on the values of its parameters A and W .

A different, but related, problem is generalizing the FFT or the IFFT to nonequispaced sampling points on the unit circle¹⁹. This problem is solved with approximate or iterative algorithms^{19–22}. The accuracy and speed of approximate algorithms depend on the desired precision²⁰, which is often controlled by an oversampling

parameter²². The computational complexity of iterative algorithms^{19,22} depends on the condition number of the problem, i.e., they may require many iterations to converge. In contrast, the CZT and ICZT algorithms are exact and their computational complexity depends only on the problem size.

Summary of the CZT and ICZT Algorithms

Forward CZT. The chirp z-transform (CZT) is defined⁷ as follows:

$$X_k = \sum_{j=0}^{N-1} x_j A^{-j} W^{jk}, \quad k = 0, 1, \dots, M-1, \quad (1)$$

where $\mathbf{x}$ is the input vector of length N and $\mathbf{X}$ is the output vector of length M . Using matrix notation this can be stated as:

$$\mathbf{X} = \mathbf{W} \mathbf{A} \mathbf{x}. \quad (2)$$

In this formula, $\mathbf{A}$ is the following diagonal matrix

$$\mathbf{A} = \text{diag}(A^{-0}, A^{-1}, \dots, A^{-(N-1)}) \quad (3)$$

and $\mathbf{W}$ is the following Vandermonde matrix:

$$\mathbf{W} = \begin{bmatrix} W^{0 \cdot 0} & W^{1 \cdot 0} & \dots & W^{(N-1) \cdot 0} \\ W^{0 \cdot 1} & W^{1 \cdot 1} & \dots & W^{(N-1) \cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ W^{0 \cdot (M-1)} & W^{1 \cdot (M-1)} & \dots & W^{(N-1) \cdot (M-1)} \end{bmatrix}. \quad (4)$$

Using Bluestein's substitution⁹, i.e., $jk = (j^2 + k^2 - (k-j)^2)/2$, the matrix $\mathbf{W}$ can be expressed as $\mathbf{W} = \mathbf{P} \hat{\mathbf{W}} \mathbf{Q}$. That is, it is equal to a product of three matrices where $\hat{\mathbf{W}}$ is the following Toeplitz matrix

$$\hat{\mathbf{W}} = \begin{bmatrix} W^{-\frac{(0-0)^2}{2}} & W^{-\frac{(0-1)^2}{2}} & \dots & W^{-\frac{(0-(N-1))^2}{2}} \\ W^{-\frac{(1-0)^2}{2}} & W^{-\frac{(1-1)^2}{2}} & \dots & W^{-\frac{(1-(N-1))^2}{2}} \\ \vdots & \vdots & \ddots & \vdots \\ W^{-\frac{((M-1)-0)^2}{2}} & W^{-\frac{((M-1)-1)^2}{2}} & \dots & W^{-\frac{((M-1)-(N-1))^2}{2}} \end{bmatrix} \quad (5)$$

and $\mathbf{P}$ and $\mathbf{Q}$ are the following two diagonal matrices:

$$\mathbf{P} = \text{diag} \left(W^{\frac{0^2}{2}}, W^{\frac{1^2}{2}}, \dots, W^{\frac{(M-1)^2}{2}} \right) \quad \text{and} \quad \mathbf{Q} = \text{diag} \left(W^{\frac{0^2}{2}}, W^{\frac{1^2}{2}}, \dots, W^{\frac{(N-1)^2}{2}} \right). \quad (6)$$

Thus, the CZT algorithm can be viewed as an efficient implementation of the following matrix equation:

$$\mathbf{X} = \mathbf{P} (\hat{\mathbf{W}} (\mathbf{Q} (\mathbf{A} \mathbf{x}))). \quad (7)$$

By exploiting the structure of the matrices, the output vector $\mathbf{X}$ can be computed in $O(n \log n)$ time, where $n = \max(M, N)$.

Algorithm S5 in Supplementary Section S2 gives the pseudo-code for the forward CZT.

Inverse CZT. In the square case, i.e., when $M = N$, the ICZT can be stated¹⁰ by inverting Eq. (7), which leads to:

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{Q}^{-1} \hat{\mathbf{W}}^{-1} \mathbf{P}^{-1} \mathbf{X}. \quad (8)$$

Because $\mathbf{A}$, $\mathbf{Q}$, and $\mathbf{P}$ are diagonal matrices, it is straightforward to compute the inverse matrices $\mathbf{A}^{-1}$, $\mathbf{Q}^{-1}$, and $\mathbf{P}^{-1}$. The symmetric Toeplitz matrix $\hat{\mathbf{W}}$ can be inverted¹⁰ using a special case of the Gohberg–Semencul formula^{23,24}. In other words, the inverse matrix $\hat{\mathbf{W}}^{-1}$ is given by:

$$\hat{\mathbf{W}}^{-1} = \frac{1}{u_0} (\mathcal{A} \mathcal{A}^T - \mathcal{D}^T \mathcal{D}), \quad (9)$$

where $\mathcal{A}$ is a lower-triangular Toeplitz matrix and $\mathcal{D}$ is an upper-triangular Toeplitz matrix. Both $\mathcal{A}$ and $\mathcal{D}$ are defined by the same generating vector $\mathbf{u} = (u_0, u_1, u_2, \dots, u_{n-1})$, i.e.,

$$\mathcal{A} = \begin{bmatrix} u_0 & 0 & 0 & \dots & 0 \\ u_1 & u_0 & 0 & \dots & 0 \\ u_2 & u_1 & u_0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ u_{n-1} & u_{n-2} & u_{n-3} & \dots & u_0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} 0 & u_{n-1} & u_{n-2} & \dots & u_1 \\ 0 & 0 & u_{n-1} & \dots & u_2 \\ 0 & 0 & 0 & \dots & u_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}. \quad (10)$$

The value of u_k for each $k \in \{0, 1, \dots, n-1\}$ is given by:

$$u_k = (-1)^k \frac{W^{\frac{2k^2 - (2n-1)k + n(n-1)}{2}}}{\prod_{s=1}^{n-k-1} (W^s - 1) \prod_{s=1}^k (W^s - 1)}. \quad (11)$$

Combining these results leads to the following closed-form solution for the ICZT:

$$\mathbf{x} = \frac{1}{u_0} \mathbf{A}^{-1} \mathbf{Q}^{-1} (\mathcal{A} \mathcal{A}^T - \mathcal{D}^T \mathcal{D}) \mathbf{P}^{-1} \mathbf{X}. \quad (12)$$

Algorithm 1 computes this expression in $O(n \log n)$ time by exploiting the structure of the matrices. See reference¹⁰ for more details, proofs, and the pseudo-code for TOEPLITZMULTIPLY and all of its dependencies.

Algorithm 1. ICZT algorithm. Runs in $O(n \log n)$ time.

```

1: ICZT( $\mathbf{X}$ ,  $N$ ,  $W$ ,  $A$ )
2:  $M \leftarrow \text{LENGTH}(\mathbf{X})$ ;
3: if  $M \neq N$  then
4:   ERROR("M must be equal to N.");
5: end if
6:  $n \leftarrow N$ ;
7:  $\mathbf{x} \leftarrow \text{EMPTYARRAY}(n)$ ;
8: for  $k \leftarrow 0$  to  $n-1$  do
9:    $x[k] \leftarrow W^{-\frac{k^2}{2}} \cdot X[k]$ ; // multiply  $\mathbf{P}^{-1}$  and  $\mathbf{X}$ 
10: end for
11: // Precompute the necessary polynomial products.
12:  $\mathbf{p} \leftarrow \text{EMPTYARRAY}(n)$ ;
13:  $p[0] \leftarrow 1$ ;
14: for  $k \leftarrow 1$  to  $n-1$  do
15:    $p[k] \leftarrow p[k-1] \cdot (W^k - 1)$ ;
16: end for
17: // Compute the generating vector  $\mathbf{u}$ .
18:  $\mathbf{u} \leftarrow \text{EMPTYARRAY}(n)$ ;
19: for  $k \leftarrow 0$  to  $n-1$  do
20:    $u[k] \leftarrow (-1)^k \frac{W^{\frac{2k^2 - (2n-1)k + n(n-1)}{2}}}{p[n-k-1] \cdot p[k]}$ ;
21: end for
22:  $\mathbf{z} \leftarrow \text{ZEROVECTOR}(n)$ ; // vector with  $n$  zeros
23:  $\hat{\mathbf{u}} \leftarrow (0, u[n-1], u[n-2], \dots, u[2], u[1])$ ;
24:  $\tilde{\mathbf{u}} \leftarrow (u[0], \underbrace{0, 0, \dots, 0}_{n-1 \text{ zeros}})$ ;
25:  $\mathbf{x}' \leftarrow \text{TOEPLITZMULTIPLY}(\hat{\mathbf{u}}, \mathbf{z}, \mathbf{x})$ ; //  $\mathcal{D}$ 
26:  $\mathbf{x}' \leftarrow \text{TOEPLITZMULTIPLY}(\mathbf{z}, \hat{\mathbf{u}}, \mathbf{x}')$ ; //  $\mathcal{D}^T$ 
27:  $\mathbf{x}'' \leftarrow \text{TOEPLITZMULTIPLY}(\mathbf{u}, \tilde{\mathbf{u}}, \mathbf{x})$ ; //  $\mathcal{A}^T$ 
28:  $\mathbf{x}'' \leftarrow \text{TOEPLITZMULTIPLY}(\tilde{\mathbf{u}}, \mathbf{u}, \mathbf{x}'')$ ; //  $\mathcal{A}$ 
29: for  $k \leftarrow 0$  to  $n-1$  do
30:    $x[k] \leftarrow \frac{x''[k] - x'[k]}{u[0]}$ ; // subtract and divide by  $u_0$ 
31: end for
32: for  $k \leftarrow 0$  to  $n-1$  do
33:    $x[k] \leftarrow A^k \cdot W^{-\frac{k^2}{2}} \cdot x[k]$ ; // multiply by  $\mathbf{A}^{-1} \mathbf{Q}^{-1}$ 
34: end for
35: return  $\mathbf{x}$ ;

```

Farey sequences and ICZT singularities

This section proves that the singularities of the ICZT on the unit circle are related to the elements of the Farey sequence of order $n - 1$, where n is the size of the transform.

Definition 1. A Farey sequence of order n is denoted by F_n . It consists of all rational numbers in the interval $[0, 1]$ with an irreducible fraction representation p/q that satisfies the inequality $q \leq n$, where n is a positive integer.

A fraction p/q is irreducible if there is no other fraction a/b for which $|a| < |p|$ or $|b| < |q|$ such that $p/q = a/b$. For example, $0/1$ is irreducible, but $0/5$ is not. To give another example, both $2/4$ and $3/6$ are reducible to $1/2$.

By mathematical convention, the numbers in each Farey sequence are sorted in increasing order. For example, the first five Farey sequences are equal to:

$$\begin{aligned} F_1 &= \left(\frac{0}{1}, \frac{1}{1} \right), \\ F_2 &= \left(\frac{0}{1}, \frac{1}{2}, \frac{1}{1} \right), \\ F_3 &= \left(\frac{0}{1}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1} \right), \\ F_4 &= \left(\frac{0}{1}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \frac{1}{1} \right), \\ F_5 &= \left(\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1} \right). \end{aligned}$$

The Farey sequence of order n contains all elements of the Farey sequence of order $n - 1$ and some new elements that are unique to F_n . Because F_n includes all elements of F_{n-1} and F_{n-1} includes all elements of F_{n-2} , it follows that F_n includes all elements of F_{n-2} as well. In fact, F_n includes all elements of $F_1, F_2, \dots, F_{n-1}$. This property becomes more clear if the Farey sequences shown above are rewritten with extra space between some of the terms as shown below:

$$\begin{aligned} F_1 &= \left(\frac{0}{1}, \quad \quad \quad \frac{1}{1} \right), \\ F_2 &= \left(\frac{0}{1}, \quad \quad \quad \frac{1}{2}, \quad \quad \quad \frac{1}{1} \right), \\ F_3 &= \left(\frac{0}{1}, \quad \quad \frac{1}{3}, \quad \quad \frac{1}{2}, \quad \quad \frac{2}{3}, \quad \quad \frac{1}{1} \right), \\ F_4 &= \left(\frac{0}{1}, \quad \frac{1}{4}, \frac{1}{3}, \quad \frac{1}{2}, \quad \frac{2}{3}, \frac{3}{4}, \quad \frac{1}{1} \right), \\ F_5 &= \left(\frac{0}{1}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{2}{3}, \frac{3}{4}, \frac{4}{5}, \frac{1}{1} \right). \end{aligned}$$

This visualization also illustrates another property of Farey sequences: the number of times that a fraction $p/q \in F_n$ appears in the sequences $F_1, F_2, \dots, F_{n-1}$ is equal to $n - q$. For example, the fraction $1/3$ appears $5 - 3 = 2$ times in F_1, F_2, F_3 , and F_4 .

The next definition links Farey fractions to polar angles.

Definition 2. Each angle θ that can be expressed in radians as $\theta = 2\pi p/q$ is a Farey angle of order n if p/q is an irreducible fraction that is an element of the Farey sequence of order n .

Note that each Farey sequence includes two distinct elements $0/1$ and $1/1$. However, these two fractions map to the same Farey angle, i.e., 0 and 2π are equivalent.

The following theorem proves that for each Farey angle the corresponding complex exponential $e^{i2\pi p/q}$ is a root of unity of order q . The converse is also true: each root of unity corresponds to a Farey angle and the order of the root of unity determines the order of the Farey sequence in which the fraction p/q first appears.

Theorem 1. Let $\theta \in [0, 2\pi)$ be an angle expressed in radians. Then, θ is a Farey angle of order n if and only if there is an integer $q \in \{1, 2, \dots, n\}$ such that the complex exponential $e^{i\theta}$ is a root of unity of order q for some $q \leq n$.

Proof. ($\Rightarrow$) Suppose that θ is a Farey angle of order n . Then, $\theta = 2\pi p/q$, where $p/q \in F_n$. This implies that the complex exponential $e^{i\theta}$ is a root of unity of order q . That is,

$$(e^{i\theta})^q = \left(e^{i2\pi \frac{p}{q}} \right)^q = e^{i2\pi p} = e^{i2\pi p} = 1, \quad (13)$$

because p is an integer.

($\Leftarrow$) Suppose that the complex exponential $e^{i\theta}$ is a root of unity of order $q \leq n$, i.e.,

$$(e^{i\theta})^q = e^{i\theta q} = 1. \quad (14)$$

Figure 3. Three chiral contours with 16 points on the unit circle. The contour points are numbered in increasing order to illustrate the cases when a contour completes more than one revolution over the unit circle. From left to right, the polar angle of the parameter W is: 22.5° , 49.0° , and 76.0° .

This equation implies that θq is an integer multiple of 2π . That is, without loss of generality, $\theta q = 2\pi p$, where p is a non-negative integer between 0 and $q - 1$. Therefore,

$$\theta = \frac{2\pi p}{q}, \quad (15)$$

which implies that $\theta \in F_q \subseteq F_p$, as required. $\square$

The next theorem ties the Farey angles that correspond to the elements of F_{n-1} to the singularities of Eq. (11), which defines the generating vector $\mathbf{u}$ that is used in the closed-form expression for the inverse chiral z-transform.

Theorem 2. Let $p/q \in \mathbb{Q}$ be an irreducible fraction in the interval $[0, 1]$. Let n be a positive integer and let $W = e^{i2\pi p/q}$. If the fraction p/q is an element of the Farey sequence of order $n - 1$, i.e., $p/q \in F_{n-1}$, then there is at least one element of the vector $\mathbf{u}$ for which the denominator in Eq. (11) is zero.

Conversely, if $p/q \notin F_{n-1}$, then all elements of the vector $\mathbf{u}$ defined by Eq. (11) have finite magnitudes and are well-defined.

Proof. Because $|W| = 1$, the absolute value of the numerator in Eq. (11) is equal to 1 for each $k \in \{0, 1, 2, \dots, n - 1\}$.

($\Rightarrow$) Suppose that $p/q \in F_{n-1}$. Then, by definition, $q \leq n - 1$. Therefore, there is at least one zero term in the product that defines the denominator of the element u_0 . More formally,

$$\prod_{s=1}^{n-1} (W^s - 1) = \prod_{s=1, s \neq q}^{n-1} (W^s - 1) \prod_{s=q}^q (W^s - 1) = \prod_{s=1, s \neq q}^{n-1} \left(e^{i2\pi \frac{ps}{q}} - 1 \right) \underbrace{\left(e^{i2\pi \frac{pq}{q}} - 1 \right)}_0 = 0. \quad (16)$$

($\Leftarrow$) Conversely, suppose that $p/q \notin F_{n-1}$. Then, $q \geq n$. Because p and q are coprime, this implies that $ps/q \notin \mathbb{Z}$ for each $s \in \{1, 2, \dots, n - 1\}$. Therefore, each term in the denominator of each element u_k is non-zero. $\square$

Supplementary Section S4 gives additional proofs and examples that perform the analysis using the rank of the Vandermonde matrix W in Eq. (4) instead of the elements of the generating vector $\mathbf{u}$, which is defined by Eq. (11). In other words, the proof in this section is based on the singularities of the closed-form formula for the ICZT, which is implemented by the ICZT algorithm. The proof in Supplementary Section S4 is based on the rank of the ICZT transformation matrix. Because the two sets of singularities are the same, any algorithm that computes or approximates the ICZT on the unit circle will have singularities at Farey angles that correspond to the elements of F_{n-1} , where n is the transform size. That is, the singularities at Farey angles are inherent to the mathematical problem; they are not introduced by Algorithm 1.

Supplementary Section S6 includes plots of the condition numbers for the transform matrix W for chiral contours with 16 and 32 points. The results show that the condition number also spikes near Farey angles. The shapes of the condition number plots are similar to the shapes of the numerical error plots. The condition numbers were computed from the singular value decomposition of the matrix W . This was done using the following formula: $\text{cond}(W) = \sigma_{\max}/\sigma_{\min}$, where $\sigma_{\max}$ is the largest singular value and $\sigma_{\min}$ is the smallest singular value of the matrix W .

Results

Results for 16-point contours. This subsection summarizes the results of the first experiment, which studied the properties of the numerical error for circular chiral contours with 16 points. The number of points is too small for most practical applications, but the lessons learned from these contours allowed us to derive the error prediction formulas described later in the paper.

Figure 3 shows three different chiral contours that lie on the unit circle. Each contour has 16 points and each point is labeled with its index, a number between 0 and 15. The leftmost contour performs exactly one revolution. The other contours perform between 2 and 3 complete revolutions over the unit circle. In all three cases, the contours are traversed clockwise because the polar angle of W is positive. The reason for this is that chiral contours are defined in terms of the z-transform, which uses negative powers (see Supplementary Section S1).

Figure 4. (a) Absolute numerical error of the CZT-ICZT procedure, as a function of the polar angle of W for 16-point chirp contours on the unit circle. The discretization step was 0.1° , i.e., there were 3600 angles. The three red points indicate the error for the three contours shown in Fig. 3. (b) The plot from (a) with the error prediction from Eq. (21) superimposed in red.

Figure 5. (a) Close-up of the harmonic hedgehog located between the first two red points in Fig. 4a. The numerical error spikes when the polar angle of W is close to an element of the set $\left\{\frac{360^\circ}{15}, \frac{360^\circ}{14}, \frac{360^\circ}{13}, \dots, \frac{360^\circ}{8}\right\}$. The eight red points indicate the numerical error for these Farey angles. (b) The plot from (a) with the numerical error estimate superimposed in red.

Figure 4a shows the absolute numerical error of the CZT-ICZT procedure (see Methods) for 16-point contours on the unit circle. Fig. 4b shows the same plot, but with the numerical error predictions superimposed in red (see Eq. (21)). The horizontal axis corresponds to the polar angles of the parameter W , which are discretized at 0.1° intervals, i.e., there are 3600 angles. The vertical axis shows the decimal logarithm of the absolute numerical error. For each of the 3600 angles, the error was averaged over 10 randomly generated input vectors after computing the logarithm. The same random vectors were used to compute the error for each point. The three red points in the plot correspond to the three chirp contours shown in Fig. 3.

Figure 5a shows a close-up view of Fig. 4a for angles between 20° and 46° . The discretization step in this case is 0.005° , which is sufficient to reveal the finer structure of the error function. The figure shows that the numerical error spikes for angles that are close to the elements of the set $\left\{\frac{360^\circ}{15}, \frac{360^\circ}{14}, \frac{360^\circ}{13}, \dots, \frac{360^\circ}{8}\right\}$. These angles form a subset of the harmonic sequence $\left(\frac{360^\circ}{1}, \frac{360^\circ}{2}, \frac{360^\circ}{3}, \dots\right)$, which is why we called this pattern of spikes the *harmonic hedgehog*. These angles correspond to elements of the Farey sequence F_{15} between $1/15$ and $1/8$. The eight red points indicate the numerical errors for these Farey angles. The red points were explicitly added to the figure using Algorithm 3, i.e., their corresponding Farey angles were added to the list of discretized polar angles of W because even the finer discretization missed them. Figure 5b shows the same plot as in Fig. 5a, but with the error predictions superimposed in red. The singularities of the transform coincide with the discontinuities of the error prediction function. This figure also shows that the spread of the points in the empirical error plot is due to numerical rounding and the residual randomness that is not fully mitigated by averaging over only 10 input vectors.

The absolute numerical error for the Farey angles shown in Fig. 5a is large, but finite. The reason why the error is bounded and not infinite as predicted by the theoretical arguments in Theorem 2 is that, in practice, the IEEE-754 floating point numbers²⁵ approximate these angles and their complex exponentials. Because π is

Figure 6. Absolute numerical error of the CZT-ICZT procedure for chirp contours with 1024 points on the unit circle. The results in (a) are for 3600 regularly-spaced angles with a discretization step of 0.1° . This discretization hits many Farey angles of lower orders, which leads to the layering of the numerical error function. See also Supplementary Fig. S3. The results in (b) are for 3600 polar angles that were randomly sampled from a uniform distribution. The random sampling is less likely to hit Farey angles, which explains the absence of the layering effect seen in (a).

irrational, the computational representation of all Farey angles except 0° is not exact, but approximate when they are expressed in radians. The floating-point computations use numbers that are very close, but still a tiny bit different from these Farey angles. This often leads to large, but finite, numerical errors.

Results for 32-point contours. Supplementary Section S5 shows the results for 32-point chirp contours. They are similar to the results for the 16-point chirp contours, but the error function appears to be compressed horizontally by a factor of two. The harmonic hedgehog can also be observed, but it is now squeezed in the interval $[11^\circ, 23^\circ]$ instead of the interval $[22^\circ, 46^\circ]$ and has twice as many spikes. In this case, the ICZT singularities correspond to the Farey sequence F_{31} instead of F_{15} . Supplementary Section S6 also includes plots of the condition number of the transform matrix W for 32-point contours.

Results for 1024-point contours. The next set of experiments investigated the behavior of the numerical error of the CZT-ICZT procedure for contours with 1024 points on the unit circle. The results show that even for large transform sizes there are many values of W for which the ICZT can be computed accurately. These experiments also studied how the plot of the error function is affected by the discretization step for the polar angles of W .

Figure 6a shows the absolute numerical error for 3600 polar angles of the transform parameter W . These angles were selected using a regularly-spaced sampling grid with a step of 0.1° . For each angle, the numerical error was averaged over 10 random input vectors (see Methods). This discretization hits many Farey angles of different orders, which leads to the stratified appearance of the plot. This layering effect is explained by the link between Farey sequences and the prime factorization of the number of regularly-discretized angles. See Supplementary Section S4 for more details.

Figure 6b shows another plot with 3600 points, also for a transform of size 1024. In this case, however, the angles were sampled at random from a uniform distribution. The resulting pattern appears to be closer to the overall shape of the error function from the previous subsections, in which the transform size was smaller. The reason for the shape difference between Fig. 6a and Fig. 6b is that the random sampling is unlikely to select a Farey angle where the ICZT is singular.

Figure 7 further explores the relationship between the error function and the discretization step. It shows nine plots of the absolute numerical error, computed for 1024-point chirp contours. These plots have completely different shapes, even though they were all computed for a transform of size 1024. In most cases, changing the number of regularly-sampled angles even a little bit leads to a different shape of the error function. Interestingly, the error function appears stratified in all six plots in the first two columns of Fig. 7, with different numbers and locations of the layers. This layering is absent in the last column of the figure, because these plots are drawn for a prime number of discretized polar angles. In all three cases, the prime number is greater than the transform size, which prevents hitting any ICZT singularities exactly.

The bottom row of Fig. 7 shows three plots that were obtained by regular sampling of the polar angles of W when the number of samples is about 4 times greater than the number of contour points. More specifically, the plots were drawn with 4096, 4098, and 4099 samples, respectively. Both 4096 and 4098 share integer factors with 1024, which explains the presence of layers in Fig. 7g,h. There are more layers in Fig. 7g compared to Fig. 7h because 1024 shares more factors with 4096 than 4098. There is no layering in Fig. 7i because 4099 and 1024 are coprime. Nevertheless, for some angles in Fig. 7i, the numerical error is high because these angles are sufficiently close to a nearby Farey angle of order less than 1024.

In other words, a plot in which the horizontal axis is discretized using a fixed step can lead to a misleading picture of the error. The behavior of the error may be completely different between the discretized points. The shape of the plot depends both on the sampling procedure and on the distribution of Farey angles. As shown below, the

Figure 7. Absolute numerical error as a function of the polar angle of the transform parameter W , plotted for chirp contours with 1024 points on the unit circle. The nine plots illustrate the variety of shapes that the error function can take depending on the choice of discretization. In all plots, the size of the transform is fixed at 1024. What varies is the number of angles, i.e., polar angles of W that are discretized using regularly-spaced intervals. Each point in each plot shows the average value of the absolute error, computed with the CZT-ICZT procedure over 10 random input vectors. The top row, i.e., plots (a–c), shows the results for the case when the number of regularly-spaced polar angles is close to the number of points on the chirp contour, i.e., 1024. The second and the third row show the results when the number of angles is approximately 2 times and 4 times greater than the size of the transform, respectively. The left column, i.e., plots (a,d,g), shows the results for the case when the number of angles is a power of two. The plots in the middle column are for discretizations with 1026, 2050, and 4098 angles, which are composite numbers that are not powers of two. The right column shows plots for the case when the number of points is a prime number.

number of Farey angles where the transform is singular grows quadratically with the size of the transform, which makes it difficult to draw a complete plot of the error function for large values of n .

As proven in Theorem 2, the singularities of the ICZT of size n on the unit circle are related to the elements of the Farey sequence of order $n - 1$. Table 1 shows that as n increases the length of the Farey sequence F_{n-1} also increases²⁶. These values imply that the number of singularities grows faster than the transform size. This growth is roughly quadratic, which follows from the popular approximation formula $|F_n| \sim \frac{3n^2}{\pi^2}$ for the length of F_n (see ref. ²⁷, p. 268 and ref. ²⁸, p. 156).

The large number of singularities makes it more difficult to draw and interpret the error plots. For example, when $N = 1024$ there are 318453 singularities. In other words, the number of singularities is approximately two orders of magnitude larger than the number of all points plotted in Fig. 6a. This explains why even a small change in the discretization of the angles may lead to a completely different numerical error profile, as illustrated in Fig. 7.

Some discretizations select more Farey angles of small orders than others. For example, the discretization used in Fig. 6a hits many of these singularities and leads to the layering of the error function (see also Fig. S3, which colors each point based on its corresponding Farey order). The discretizations with prime number of angles shown in Fig. 7c,f,i don't hit any small-order Farey angles exactly. This is true for any regular discretization in

n	16	32	64	128	256	512	1024	2048
$ F_{n-1} $	73	309	1,229	4,959	19,821	79,597	318,453	1,274,563

Table 1. Length of the Farey sequence F_{n-1} as a function of n . The ICZT of size n has $|F_{n-1}|$ singularities when $|W| = 1$.

Figure 8. Absolute numerical error for the CZT-ICZT procedure, computed with two different floating-point precisions. In both plots, the results are for 2048-point contours on the unit circle and for 4099 polar angles of W that were discretized using regular intervals. The results in (a) were computed with double-precision, which is implemented natively in modern CPUs (i.e., 64-bit numbers in IEEE-754 format)²⁵. The results in (b) were computed with 128-bit floating-point numbers, i.e., quadruple precision, implemented by GCC's *libquadmath* library³³ through an interface in *Boost Multiprecision*³⁴.

which the prime number is greater than or equal to the transform size. The next two subsections use discretizations with 4099 angles, which is a large prime number.

Results for 2048-point contours. The next experiment studied the behavior of the error function for contours with 2048 points. Our previous experience with the ICZT indicated that the numerical accuracy decreases for contours with large number of points¹⁰. This is consistent with previous observations that Vandermonde systems can be ill-conditioned and should be solved with double precision or higher²⁹. Interestingly, it was also shown that some ill-conditioned Vandermonde systems can be solved with high precision^{30–32}. In our previous work, however, the chirp contours were off the unit circle, i.e., they were expanding or contracting logarithmic spiral contours. We wanted to check if additional numerical precision is still needed for contours with thousands of points that are restricted to lie on the unit circle.

Figure 8 shows the absolute numerical error of the CZT-ICZT procedure for 2048-point contours. These results are for 4099 polar angles that were discretized using regular intervals. The two plots were computed with two different floating-point precisions. Figure 8a shows the results for double precision, which is implemented natively in modern CPUs. Figure 8b shows a different plot that was computed with quadruple precision. It was generated using the *libquadmath* library³³ through an interface provided by *Boost Multiprecision*³⁴, which makes it easier to use various floating point formats in C++. The plots show that the numerical accuracy can be boosted by increasing the precision of the floating-point numbers. Switching from double precision to quadruple precision reduces the error by approximately 16 orders of magnitude. The overall shape, however, remains mostly the same.

The results also indicate that the transforms are computable for $N = 2048$ using only double precision. Unless the polar angle of W is close to a Farey angle, the error is usually small when the chirp contour is on the unit circle. Nevertheless, if additional numerical accuracy is needed, then one can switch from double to quadruple floating-point precision, which is becoming more popular and easier to use.

Formulas for predicting the numerical error. This subsection states formulas that approximate the absolute numerical error for the sequential applications of the forward and inverse transforms, i.e., CZT followed by ICZT and vice versa. The formulas are given for the square case in which $M = N$ and for chirp contours that lie on the unit circle. The error formulas for a more general case with logarithmic spiral contours that span a 360° arc off the unit circle are given in ref. ¹⁰.

The numerical error can be modeled using five terms: U_1 , U_2 , U_3 , T , and B . The first three terms are derived from the elements of the generating vector $\mathbf{u}$ (see Eq. (11)). For contours on the unit circle they are equal to:

$$U_1 = \log \sqrt{\sum_{k=1}^{N-1} |u_k|^2} = \frac{1}{2} \log \sum_{k=1}^{N-1} |u_k|^2, \quad U_2 = \log \sqrt{\sum_{k=0}^{N-1} |u_k|^2} = \frac{1}{2} \log \sum_{k=0}^{N-1} |u_k|^2, \quad U_3 = -\log |u_0|. \quad (17)$$

The term U_1 is equal to the logarithm of the Euclidean norm of the first row of the matrix $\mathcal{D}$, which includes all elements of $\mathbf{u}$ except u_0 (see Eq. (10)). The term U_2 is equal to the logarithm of the Euclidean norm of the first

N	CZT-ICZT		ICZT-CZT	
	Avg.	Std.	Avg.	Std.
16	0.96977	3.43535×10^{-4}	0.97642	2.94905×10^{-4}
32	0.98703	8.90973×10^{-5}	0.98932	1.15295×10^{-4}
64	0.99453	3.43848×10^{-5}	0.99520	1.75565×10^{-5}
128	0.99656	7.65721×10^{-6}	0.99680	1.40965×10^{-5}
256	0.99752	5.36489×10^{-6}	0.99758	4.72754×10^{-6}
512	0.99823	3.00392×10^{-6}	0.99824	2.84512×10^{-6}
1024	0.99863	9.77751×10^{-7}	0.99863	1.89758×10^{-6}
2048	0.99871	8.76893×10^{-7}	0.99871	1.55039×10^{-6}

Table 2. R^2 fits for the predicted numerical error for regularly-sampled polar angles of the transform parameter W .

column of the matrix $\mathcal{A}$, which is defined by Eq. (10). It is similar to U_1 , but also includes u_0 in addition to all other elements. Finally, the term U_3 models the division by u_0 in Eq. (12). Thus, U_3 is equal to the logarithm of $\frac{1}{|u_0|}$, where u_0 is not equal to zero by definition. For the error formulas to be accurate, these terms must be computed using the same numerical precision that is used by the ICZT algorithm, preferably using the same version of the vector $\mathbf{u}$.

The fourth term, T , is the sum of three simpler terms¹⁰. In this special case, however, they are all equal to $\frac{1}{2}\log N$ because the transform parameters A and W have a magnitude of 1, i.e., $|A| = |W| = 1$. Thus, the term T is equal to:

$$T = \frac{3}{2}\log N. \quad (18)$$

The formulas also include an offset term, B , that depends on the size of the transform, N , and the number of precision bits, p , used in the calculations. This offset determines the base level of the error function. It is defined as follows:

$$B = -p \log 2 + C_1 \log N + C_2, \quad (19)$$

where C_1 and C_2 are implementation-dependent constants. For 64-bit and 128-bit IEEE-754 floating-point numbers²⁵, the value of p is set to 53 and 113, respectively.

Error formula for the CZT followed by the ICZT. The absolute numerical error for a sequential application of the CZT followed by the ICZT is equal to the Euclidean distance between the original input vector $\mathbf{x}$ and the computed vector $\hat{\mathbf{x}}$. That is,

$$E = \|\hat{\mathbf{x}} - \mathbf{x}\|. \quad (20)$$

The error values for the CZT-ICZT procedure can be analytically approximated using the following formula:

$$\log E \approx U_1 + U_2 + U_3 + T + B + \log \|\mathbf{x}\|. \quad (21)$$

Error formula for the ICZT followed by the CZT. The absolute error of the ICZT-CZT procedure is

$$E = \|\hat{\mathbf{X}} - \mathbf{X}\|, \quad (22)$$

where $\mathbf{X}$ is the true output vector and $\hat{\mathbf{X}}$ is the computed output vector. The log of the error is approximately equal to:

$$\log E \approx U_1 + U_2 + U_3 + T + B + \log \|\mathbf{X}\|. \quad (23)$$

Evaluation of the Error Prediction Formulas. The accuracies of Eqs. (21) and (23) were evaluated using the two procedures described above. The difference between predicted and empirically observed errors was quantified using the R^2 coefficient (see Methods). Table 2 shows the means and the standard deviations of the R^2 coefficients for the two experimental procedures and for transform sizes between 16 and 2048. In all cases, the results were computed by averaging the R^2 values from 10 independent runs of the corresponding procedure, where each run used 10 random input vectors. For each value of N , the fits used 4099 regularly-discretized polar angles of W . The average R^2 value increases with the size of the transform and approaches 1, which indicates that the formulas predict the numerical error very well. The results also show that there is no substantial difference between the R^2 values for the two experimental procedures.

Additional results for the accuracy of the error prediction formulas are given in Supplementary Sections S7 and S8.

Figure 9. Absolute numerical error for 16-point chirp contours, shown as a function of the polar angle of W for two different experimental procedures: (a) the CZT followed by Algorithm 1, which implements the ICZT (this is the same as Fig. 2); and (b) the CZT followed by Algorithm 2, which uses the contour reversal approach proposed in reference¹³. The angles were discretized with a step of $360^\circ/2048$. The three red points in both (a) and (b) correspond to the chirp contours shown in Fig. 1. Each point in both plots represents the average error for 10 randomly generated unit-length input vectors.

Figure 10. Visualization of the accuracy of two experimental procedures for 16-point chirp contours. This is similar to Fig. 9, but in this case the discretization step for the polar angle of W was set to 0.1° . The three red points in both plots correspond to the chirp contours shown in Fig. 3. The plot in (a) is the same as in Fig. 4a.

Debunking the Reverse-as-Inverse Approach for Inverting the CZT on the Unit Circle

This section shows that a naïve but surprisingly popular approach for inverting the CZT by reversing the direction of the chirp contour is incorrect. The essence of this reverse-as-inverse approach is captured by a formula from ref. ¹³ that is reproduced below:

$$\text{ICZT}(X(k)) = [\text{CZT}(X(k)^*)]^*. \quad (24)$$

The formula attempts to express the ICZT of the vector $X(k)$ by conjugating all elements of the CZT of the vector $X(k)^*$, where $*$ denotes elementwise conjugation. In matrix form, Eq. (24) can be stated as follows:

$$\mathbf{y} = \overline{\mathbf{W}} \mathbf{A} \overline{\mathbf{X}} = \overline{\mathbf{W}} \overline{\mathbf{A}} \mathbf{X}. \quad (25)$$

In this formula, $\mathbf{y}$ is the resulting vector, $\overline{\mathbf{W}}$ is the matrix obtained by conjugating all elements of the Vandermonde matrix $\mathbf{W}$ from Eq. (4), $\overline{\mathbf{A}}$ is the matrix obtained by conjugating all elements of the diagonal matrix $\mathbf{A}$ from Eq. (3), and $\mathbf{X}$ is the ICZT input vector. Instead of conjugating all elements of the two matrices, it is possible to compute the same result using the conjugates of the transform parameters $\mathbf{A}$ and $\mathbf{W}$. That is, the vector $\mathbf{y}$ in Eq. (25) can be expressed as:

$$\mathbf{y} = \text{CZT}(\mathbf{X}, M, \overline{\mathbf{W}}, \overline{\mathbf{A}}). \quad (26)$$

If the chirp contour lies on the unit circle, then this formula reflects its starting point with respect to the real axis and also reverses its winding direction.

Figure 11. Visualization of the accuracy of two experimental procedures for 2048-point chirp contours. This is similar to Fig. 10, but in this case the discretization used 4099 regularly-spaced polar angles of W and the results were computed using quadruple precision (i.e., 128 bits) instead of double precision (i.e., 64 bits). The plot in (a) is the same as in Fig. 8b.

Ref. ¹³ also states that Eq. (24) needs to be interpreted “to within a scaling factor” that was not specified. That is, the equal sign denotes proportionality instead of equality. For unit-length input vectors $\mathbf{x}$ in the CZT–ICZT procedure (see Methods), optimal scaling for this approach can be achieved by normalizing the vector $\mathbf{y}$, i.e.,

$$\hat{\mathbf{x}} = \frac{\mathbf{y}}{\|\mathbf{y}\|} \quad (27)$$

This ensures that $\|\mathbf{x}\| = \|\hat{\mathbf{x}}\| = 1$, i.e., the norm of the input vector is always equal to the norm of the output vector. Algorithm 2 implements the reverse-as-inverse approach, i.e., Eq. (26) with the scaling factor from Eq. (27). For error plots on the log scale, the triangle inequality implies that $\log_{10} \|\mathbf{x} - \hat{\mathbf{x}}\| \leq \log_{10} 2 \approx 0.3$, i.e., the absolute numerical error for the CZT followed by Algorithm 2 can never exceed 0.3 on the log scale. Thus, this scaling is favorable to this algorithm. Unfortunately, the results show that the algorithm has a systematic error that cannot be corrected by scaling.

The rest of this section describes three experiments that compared the accuracy of Algorithm 1 to the accuracy of Algorithm 2. In all experiments, the value of A was equal to 1, the value of M was equal to N , and the polar angle of W was sampled using regular discretization as described in Methods. The accuracies were measured with the CZT–ICZT procedure, also described in Methods, but with Algorithm 2 replacing the ICZT implementation in the second condition.

Figure 9 shows the results of the first experiment, which used circular chirp contours with 16 points. Figure 9a, which is a copy of Fig. 2, shows that Algorithm 1 accurately inverts the CZT for many polar angles of W . In contrast, Fig. 9b shows that the reverse-as-inverse approach is accurate only when this angle is equal to -22.5° or 22.5° . For all other angles the error is consistently and unacceptably high, i.e., the absolute numerical error (before the log) is close to 1, which is the norm of the input vector $\mathbf{x}$. In both plots, the three red points correspond to the chirp contours shown in Fig. 1.

Figure 10 shows the results of the second experiment, which also used 16-point circular chirp contours. In this case, the discretization used 3600 regularly-sampled polar angles. Figure 10a, which is a copy of Fig. 4a, shows the error using Algorithm 1. Figure 10b shows the error using the reverse-as-inverse approach. The three red points in each plot correspond to the three chirp contours shown in Fig. 3. Figure 10b shows that with Algorithm 2 the results are accurate only for the eight angles that correspond to the eight primitive roots of unity of order 16 (see also Supplementary Section S2).

Finally, Fig. 11 shows the results of the third experiment, which used 2048-point circular chirp contours and 4099 regularly-discretized polar angles of W . In contrast to Figs. 9 and 10, which show results for double precision, the computations for this figure used quadruple precision. Figure 11a, which is a copy of Fig. 8b, shows the error using Algorithm 1. Figure 11b shows that the error using Algorithm 2 is again consistently large and proportional to $\|\mathbf{x}\|$. Because this discretization did not hit any primitive roots of unity of order 2048, there were no angles for which the reverse-as-inverse approach is accurate. The increased floating-point precision also did not help to reduce its error.

To summarize, the experiments showed that Algorithm 1 is accurate for most chirp contours and the numerical error of the CZT followed by the ICZT can be predicted using Eq. (21). In contrast, the reverse-as-inverse approach has large, systematic errors that are not affected by the numerical precision. It is accurate only when the chirp contour points coincide with the roots of unity of order N , i.e., only when the ICZT reduces to a permutation of the elements of the IFFT output vector. Algorithm 2 assumes that there is no interference between the frequency components, which is true only when they are orthogonal or, equivalently, when the value of W is a primitive root of unity of order N . For all other values of W the reverse-as-inverse approach fails to compute the ICZT.

Discussion

This paper generalized the inverse fast Fourier transform (IFFT) to work with contours that perform partial or multiple revolutions on the unit circle. This was accomplished by analyzing the numerical error properties of the ICZT algorithm for a special case with circular chirp contours on the unit circle. The paper also proved that the ICZT singularities are tied to elements of Farey sequences, where the Farey order is smaller than the transform size. Formulas for predicting the numerical error were derived as well. The experiments showed that these formulas fit the empirically-observed numerical errors very well.

The FFT and the IFFT are restricted to use orthogonal, harmonically-spaced frequency components that are generated by the integer powers of the complex roots of unity. The CZT and the ICZT can use non-orthogonal frequency components, even for the special case with chirp contours on the unit circle. This additional flexibility, however, comes at a cost. For some values of the transform parameter W the ICZT could be very inaccurate. The error prediction formulas derived in this paper allow the practitioners to avoid these singularities. For most values of W , however, the ICZT is accurate and can be computed for large transform sizes.

The *Chirp Transform Algorithm* (CTA) and the *Fractional Fourier Transform* (FRFT) are two popular algorithms^{15,16} that can be viewed as special cases of the CZT for chirp contours on the unit circle. The corresponding inverse algorithms, however, have not been described until now. Supplementary Section S3 states two special cases of the ICZT algorithm that invert the CTA and the FRFT. We named these special cases the *Inverse Chirp Transform Algorithm* (ICTA) and the *Inverse Fractional Fourier Transform* (IFRFT).

This paper analyzed the numerical error of the ICZT algorithm for chirp contours on the unit circle that can perform partial or multiple revolutions. The numerical error for logarithmic spiral contours off the unit circle that span 360° was analyzed in our previous paper¹⁰. Future work could combine the insights from these two studies to derive error prediction formulas for logarithmic spiral contours off the unit circle that perform partial or multiple revolutions. In that case, the frequency components are not orthogonal and can also decay or grow exponentially. Future work could also try to derive bounds for the condition number of the transformation matrix and relate them to the error formulas derived in this paper.

Methods

This section describes the methods that were used to evaluate the ICZT algorithm for chirp contours that lie on the unit circle. The experiments systematically varied the size of the transform and the sampling procedure for the polar angle of the parameter W . The transform parameter A was always set to 1 because changing its polar angle is equivalent to changing the polar angles of the elements of the ICZT output vector¹⁰, which does not affect the expected numerical error over all possible input vectors.

CZT–ICZT procedure. The experiments measured the absolute numerical error for the sequential application of the CZT followed by the ICZT. The following five steps were repeated ten times for each value of the transform parameter W : (1) generate a complex input vector $\mathbf{x}$ by sampling its real and complex parts from a uniform distribution on the interval $[-1, 1]$; (2) normalize the vector $\mathbf{x}$ so that its length is equal to 1; (3) use the vector $\mathbf{x}$ as input for the CZT algorithm, which results in the output vector $\hat{\mathbf{X}}$; (4) use the ICZT algorithm with the vector $\hat{\mathbf{X}}$ as input to compute the vector $\hat{\mathbf{x}}$; and (5) compute the Euclidean distance between the vector $\hat{\mathbf{x}}$ and the vector $\mathbf{x}$. A different random seed was used for each repetition. The absolute numerical error of the CZT–ICZT procedure was set to the average Euclidean distance over these 10 repetitions. For experiments that varied the polar angle of W , the same 10 input vectors were used for all angles, i.e., the vectors were generated once and then re-used.

ICZT–CZT procedure. Some of the experiments also used the ICZT–CZT procedure, in which the output of the ICZT algorithm was used as an input to the CZT algorithm. This procedure also consisted of five steps that were repeated ten times for each value of the transform parameter W : (1) generate a complex input vector $\mathbf{X}$ by sampling the real and complex parts of each of its elements from a uniform distribution on $[-1, 1]$; (2) normalize the vector $\mathbf{X}$ so that it has unit length; (3) use the normalized vector $\mathbf{X}$ as input for the ICZT algorithm to compute the vector $\hat{\mathbf{x}}$; (4) use the vector $\hat{\mathbf{x}}$ as input for the CZT algorithm to compute the vector $\hat{\mathbf{X}}$; and (5) compute the Euclidean distance between the vectors $\mathbf{X}$ and $\hat{\mathbf{X}}$. A different random seed was used for each repetition. The absolute numerical error of this procedure was equal to the average Euclidean distance for these 10 repetitions. For experiments that varied the value of W , the input vectors were re-used as described above.

Numerical error reporting. All numerical errors in this paper are reported on the log scale. For each polar angle of W , the Euclidean distance was averaged over 10 random input vectors after computing the logarithm. The experiments were designed such that the magnitude of the input vector and the magnitude of the expected output vector were both equal to 1. The results become exponentially more accurate as the error value decreases. For example, if the decimal logarithm of the absolute numerical error is -5 , then the Euclidean distance between the result vector and the expected output vector is equal to 0.00001.

Floating-point precisions. The experiments used two different floating-point precisions: (1) double-precision with 64 bits and (2) quadruple-precision with 128 bits. The double precision is implemented natively by modern CPUs. The quadruple precision was implemented using GCC's *libquadmath*³³. We used the interface from the *boost multiprecision* library³⁴. In both cases the floating-point numbers were stored in IEEE-754 format²⁵.

Sampling of the polar angles of the transform parameter W . Two methods were used to sample the polar angles of W : (1) regularly-spaced sampling; and (2) random sampling from a uniform distribution. Both methods sampled P angles from the interval $[0^\circ, 360^\circ)$. The figures, however, plot the angles between -180° and 180° because the largest peak of the error is centered at 0° , i.e., the interval $[180^\circ, 360^\circ)$ is plotted as negative angles in the interval $[-180^\circ, 0^\circ)$.

The regular sampling method used the following formula to select P polar angles $\theta_0, \theta_1, \theta_2, \dots, \theta_{P-1}$ to be evaluated:

$$\theta_k = 2\pi k/P, \quad \text{where } k \in \{0, 1, 2, \dots, P-1\}. \quad (28)$$

The random sampling method selected the polar angles by drawing i.i.d. samples from a uniform distribution on $[0, 2\pi)$.

Given an angle θ in radians, the transform parameter W was computed using Euler's formula: $W = e^{i\theta} = \cos\theta + i \sin\theta$.

Reverse-as-Inverse algorithm. Algorithm 2 implements Eq. (24), which attempts to invert the CZT by reversing the direction of the chirp contour. As described in the paper, this approach does not really work.

Algorithm 2. Reverse-as-Inverse algorithm with scaling.

```

1: REVERSE-AS-INVERSE( $\mathbf{X}, N, W, A$ )
2:  $\mathbf{y} \leftarrow \text{CZT}(\mathbf{X}, N, \overline{W}, \overline{A})$ ; // reverse the chirp contour
3:  $d \leftarrow 0$ ;
4: for  $k \leftarrow 0$  to  $N-1$  do
5:    $d \leftarrow d + |y[k]|^2$ ;
6: end for
7:  $\hat{\mathbf{x}} \leftarrow \text{EMPTYVECTOR}(N)$ ;
8: for  $k \leftarrow 0$  to  $N-1$  do
9:    $\hat{x}[k] \leftarrow y[k]/\sqrt{d}$ ;
10: end for
11: return  $\hat{\mathbf{x}}$ ;

```

Enumerating Farey sequences. Farey sequences^{28,35–37} are related to the singularities of the ICZT when it is computed for circular chirp contours on the unit circle. In other words, there is a connection between addition of rational numbers and multiplication of complex numbers that lie on the unit circle. Each Farey sequence F_n is formed by all irreducible fractions $p/q \in [0, 1]$, where $q \leq n$. All Farey sequences share the *mediant* property. That is, if $a/b, p/q$, and c/d are any three consecutive elements of a Farey sequence, then $p/q = (a+c)/(b+d)$.

The mediant property enables the enumeration of the elements of a Farey sequence by an algorithm³⁸. Algorithm 3 shows the pseudo-code for an efficient procedure³⁸ that generates a Farey sequence of order n . This procedure was used for some of the figures that explicitly included all ICZT singularities in a given subset of the parameter space.

Algorithm 3. Generates the Farey sequence F_n in $O(n^2)$.

```

1: LISTFAREYFRACTIONS( $n$ )
2:  $\mathbf{F} \leftarrow \text{EMPTYLIST}()$ ;
3:  $(a, b, c, d) \leftarrow (0, 1, 1, n)$ ;
4:  $\mathbf{F} \leftarrow \text{APPEND}(\mathbf{F}, a/b)$ ;
5: while  $c \leq n$  do
6:    $k \leftarrow \lfloor \frac{n+b}{d} \rfloor$ ;
7:    $(a, b, c, d) \leftarrow (c, d, kc-a, kd-b)$ ;
8:    $\mathbf{F} \leftarrow \text{APPEND}(\mathbf{F}, a/b)$ ;
9: end while
10: return  $\mathbf{F}$ ;

```

Computing the R^2 coefficient. The accuracy of the numerical error prediction formulas was evaluated using the R^2 coefficient. That is, given two vectors $\mathbf{a} = (a_0, a_1, a_2, \dots, a_{m-1})$ and $\mathbf{b} = (b_0, b_1, b_2, \dots, b_{m-1})$, the R^2 value was computed as follows:

$$R^2 = 1 - \alpha/\beta, \quad \text{where} \quad \alpha = \sum_{k=0}^{m-1} ((a_k - \bar{a}) - (b_k - \bar{b}))^2 \quad \text{and} \quad \beta = \sum_{k=0}^{m-1} (b_k - \bar{b})^2. \quad (29)$$

In this formula, $\bar{a}$ is the average of the elements in the vector $\mathbf{a}$ and $\bar{b}$ is the average of the elements in the vector $\mathbf{b}$, i.e.,

$$\bar{a} = (a_0 + a_1 + a_2 + \dots + a_{m-1})/m \quad \text{and} \quad \bar{b} = (b_0 + b_1 + b_2 + \dots + b_{m-1})/m. \quad (30)$$

In other words, the R^2 coefficient is equal to 1 minus the fraction of the variance of the vector $\mathbf{b}$ that is unexplained by the vector $\mathbf{a}$. Our approach centers the vectors (i.e., subtracts their averages) before computing R^2 . Thus, the constant offset $\bar{b} - \bar{a}$ does not affect the R^2 value.

In the experiments, the vector $\mathbf{a}$ was set to the predicted logarithms of the numerical errors. The vector $\mathbf{b}$ was set to the logarithms of the numerical errors that were computed by either the CZT-ICZT or the ICZT-CZT procedure.

Inversion of Toeplitz matrices. The CZT matrix is equal to the product of a Vandermonde matrix and a diagonal matrix. Bluestein's substitution⁹ expresses the Vandermonde matrix as the product of two diagonal matrices and a Toeplitz matrix. Thus, the key to the ICZT algorithm is finding a computationally efficient way to invert this Toeplitz matrix.

The Gohberg–Semencul formula^{23,24} expresses the inverse of a Toeplitz matrix as a difference between two products of upper-triangular and lower-triangular Toeplitz matrices. These four matrices can be described by two generating vectors $\mathbf{u}$ and $\mathbf{v}$, but the formula does not specify how to find these vectors. The matrix used by the CZT, however, is a special case of a symmetric Toeplitz matrix. For this matrix there is a special case of the Gohberg–Semencul formula with just one generating vector¹⁰. Furthermore, we were able to express the elements of this vector in terms of the transform parameter W . This led to an efficient ICZT algorithm that runs in $O(n \log n)$ time.

Efficient Toeplitz–vector multiplication. The ICZT algorithm uses fast FFT-based subroutines for multiplying a Toeplitz matrix by a vector¹⁰. There are at least two different approaches for computing these products in $O(n \log n)$ time: (1) embedding the Toeplitz matrix into a larger circulant matrix (see ref. ³⁹, p. 202) and (2) expressing the Toeplitz matrix as a sum of a circulant matrix and a skew-circulant matrix of the same shape using Pustynnikov's decomposition^{40,41}, see also ref. ⁴², p. 40 and ref. ⁴³, p. 66.

Both subroutines multiply a circulant matrix or a skew-circulant matrix⁴⁴ by a vector in $O(n \log n)$ time (i.e., fast circular convolution). The term f -circulant with $f = -1$ is also used in the literature to refer to skew-circulant matrices^{43,45}. The required FFT sizes and the order of the operations performed by these two approaches are different. Their numerical accuracy, however, is similar. In all experiments described in this paper we used the embedding approach.

Data availability

All data and procedures are described in the main paper or in the supplementary information.

Received: 3 November 2019; Accepted: 9 February 2020;

Published online: 17 March 2020

References

1. Rockmore, D. The FFT: an algorithm the whole family can use. *Computing in Science & Engineering* **2**, 60–64 (2000).
2. Dongarra, J. & Sullivan, F. Introduction to the top 10 algorithms. *Computing in Science & Engineering* **2**, 22–23 (2000).
3. Cooley, J. & Tukey, J. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* **19**, 297–301 (1965).
4. Bluestein, L. A linear filtering approach to the computation of discrete Fourier transform. In *Proceedings of the Northeast Electronic Research and Engineering Meeting (NEREM)*, 218–219 (Boston, MA, 1968).
5. Rabiner, L. & Schafer, R. The use of an FFT algorithm for signal processing. In *Proceedings of the Northeast Electronic Research and Engineering Meeting (NEREM)*, 224–225 (Boston, MA, 1968).
6. Rabiner, L., Schafer, R. & Rader, C. The chirp z-transform algorithm and its application. *Bell Systems Technical Journal* **48**, 1249–1292 (1969).
7. Rabiner, L., Schafer, R. & Rader, C. The chirp z-transform algorithm. *IEEE Transactions on Audio and Electroacoustics* **17**, 86–92 (1969).
8. Rabiner, L. The chirp z-transform – a lesson in serendipity. *IEEE Signal Processing Magazine* **21**, 118–119 (2004).
9. Bluestein, L. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio and Electroacoustics* **18**, 451–455 (1970).
10. Sukhoy, V. & Stoytchev, A. Generalizing the inverse FFT off the unit circle. *Scientific Reports* **9**, 14443 (2019).
11. Mersereau, R. *Digital reconstruction of multidimensional signals from their projections*. Ph.D. thesis, Massachusetts Institute of Technology, Chapter 5 (1973).
12. Mersereau, R. An algorithm for performing an inverse chirp z-transform. *IEEE Transactions on Acoustics, Speech and Signal Processing* **22**, 387–388 (1974).
13. Frickey, D. Using the inverse chirp-z transform for time-domain analysis of simulated radar signals. Tech. Rep., Idaho National Engineering Lab., Idaho Falls, ID (1995).
14. Gelzer, A. & Ulyanov, V. Features of chirp z-transform in a panoramic vector network analyzer in the implementation of option time domain (in Russian). *Reports of Tomsk State University of Control Systems and Radioelectronics* **24**, 162–165, Part 1 (2011).
15. Oppenheim, A. & Schafer, R. *Discrete-Time Signal Processing*, 3 edn. Chapter 9, pp. 749–750 (Prentice Hall, Upper Saddle River, NJ, 2009).
16. Bailey, D. & Swartztrauber, P. The fractional Fourier transform and applications. *SIAM Review* **33**, 389–404 (1991).

17. Swarztrauber, P., Sweet, R., Briggs, W. & Otto, J. Bluestein's FFT for arbitrary N on the hypercube. *Parallel Computing* **17**, 607–617 (1991).
18. Bostan, A. & Schost, É. Polynomial evaluation and interpolation on special sets of points. *Journal of Complexity* **21**, 420–446 (2005).
19. Dutt, A. & Rokhlin, V. Fast Fourier transforms for nonequispaced data. *SIAM Journal on Scientific Computing* **14**, 1368–1393 (1993).
20. Dutt, A. & Rokhlin, V. Fast Fourier transforms for nonequispaced data, II. *Applied and Computational Harmonic Analysis* **2**, 85–100 (1995).
21. Kircheis, M. & Potts, D. Direct inversion of the nonequispaced fast Fourier transform. *Linear Algebra and its Applications* **575**, 106–140 (2019).
22. Ruiz-Antolin, D. & Townsend, A. A nonuniform fast Fourier transform based on low rank approximation. *SIAM Journal on Scientific Computing* **40**, A529–A547 (2018).
23. Gohberg, I. & Semencul, A. On the inversion of finite Toeplitz matrices and their continuous analogs (in Russian). *Mat. issled* **2**, 201–233 (1972).
24. Gohberg, I. & Feldman, I. *Convolution Equations and Projection Methods for Their Solution (translated from Russian)*. Translations of Mathematical Monographs, Section III.6 (American Mathematical Society, Providence, RI, 1974).
25. Institute of Electrical and Electronics Engineers (IEEE). IEEE standard for floating-point arithmetic (IEEE 754-2008). Technical Standard (2008). See Table-3.5 — Binary interchange format parameters.
26. OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. Sequence A005728, <https://oeis.org/A005728> (2019).
27. Hardy, G. & Wright, E. *An Introduction to the Theory of Numbers*, 4 edn. (Oxford University Press, London, 1975).
28. Conway, J. & Guy, R. *The Book of Numbers*, corrected edn. pp. 152–156 (Copernicus, an imprint of Springer-Verlag, Inc., New York, 1996).
29. Press, W., Teukolsky, S., Vetterling, W. & Flannery, B. *Numerical Recipes*, 3 edn. p. 95 (Cambridge Univ. Press, Cambridge, MA, 2007).
30. Björck, Å & Pereyra, V. Solution of Vandermonde systems of equations. *Mathematics of Computation* **24**, 893–903 (1970).
31. Higham, N. Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems. *Numerische Mathematik* **50**, 613–632 (1987).
32. Demmel, J. & Koev, P. The accurate and efficient solution of a totally positive generalized Vandermonde linear system. *SIAM J. on Matrix Analysis and Applications* **27**, 142–152 (2005).
33. Free Software Foundation. *The GCC Quad-Precision Math Library*. <https://gcc.gnu.org/onlinedocs/gcc-8.2.0/libquadmath.pdf> (2018).
34. Maddock, J. & Kormanyos, C. *Boost Multiprecision*, https://www.boost.org/doc/libs/1_69_0/libs/multiprecision/doc/html/ (2018).
35. Farey, J. On a curious property of vulgar fractions. *The Philosophical Magazine* **47**, 385–386 (1816).
36. Niven, I. & Zuckerman, H. *An Introduction to the Theory of Numbers*, 4 edn. Chapter 6: Farey Fractions and Irrational Numbers, pp. 171–185 (Wiley, New York, 1980).
37. Graham, R., Knuth, D. & Patashnik, O. *Concrete Mathematics*, 2 edn. (Addison-Wesley, Reading, MA, 1994).
38. Routledge, N. Computing Farey series. *The Mathematical Gazette* **92**, 55–62 (2008).
39. Golub, G. & Van Loan, C. *Matrix Computations*, 3 edn. (Johns Hopkins University Press, Baltimore, MD, 1996).
40. Pustynnikov, L. On the algebraic structure of the spaces of Toeplitz and Hankel matrices (in Russian). *Soviet Math. Dokl.* **21**, 141–144 (1980).
41. Pustynnikov, L. Toeplitz and Hankel matrices and their applications (in Russian). *Uspekhi Mat. Nauk* **39**, 53–84 (1984).
42. Ng, M. *Iterative Methods for Toeplitz Systems* (Oxford University Press, Oxford, UK, 2004).
43. Pan, V. *Structured Matrices and Polynomials: Unified Superfast Algorithms* (Springer Science & Business Media, 2012).
44. Davis, P. *Circulant Matrices* (John Wiley & Sons, New York, 1979).
45. Bini, D. & Pan, V. *Polynomial and Matrix Computations. Vol. 1. Fundamental Algorithms* (Birkhäuser, Boston, MA, 1994).

Acknowledgements

We would like to thank James Oliver, the director of the Student Innovation Center and the former director of the Virtual Reality Applications Center at ISU, for creating the research environment in which we could pursue this work.

Author contributions

A.S. and V.S. designed the experiments. V.S. wrote the evaluation code and generated the tables and the figures. A.S. supervised the work. Both authors wrote the paper.

Competing interests

V.S. and A.S. are inventors on Patent Cooperation Treaty serial no. PCT/US18/43468 covering systems and methods that use the ICZT.

Additional information

Supplementary information is available for this paper at <https://doi.org/10.1038/s41598-020-60878-7>.

Correspondence and requests for materials should be addressed to A.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2020