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

Vladimir Sukhoy[1] **and Alexander Stoytchev**[1,*]

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

ABSTRACT

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.

Introduction

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 1965[3]. 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 $O(n \log n)$ time[10], 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.

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 case[13], a modified version of the forward transform, in which the circular chirp contour was traversed in the opposite direction, was

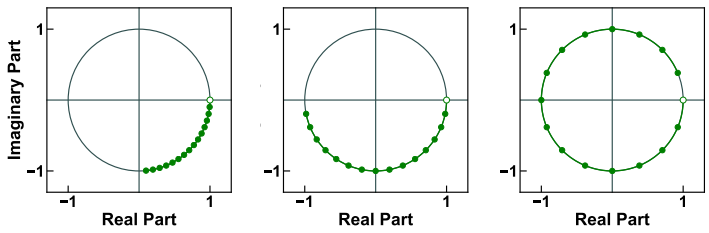

Figure 1. Three chirp contours on the unit circle, each with 16 points. The angular interval between any two adjacent points is: $90^\circ/16$, $180^\circ/16$, and $360^\circ/16$. The starting point of each contour is indicated with an unfilled circle.

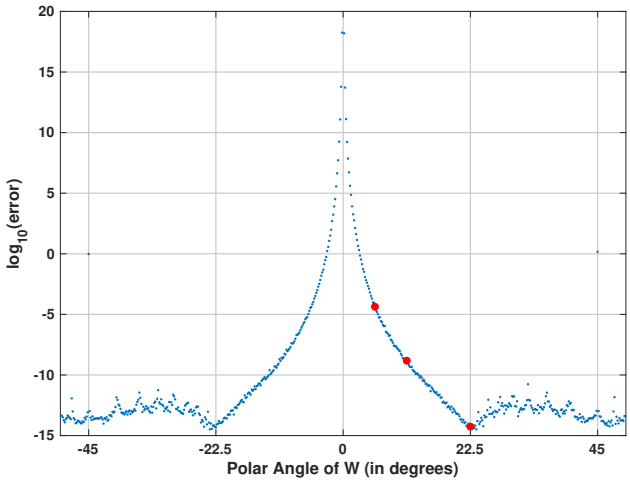

Figure 2. Absolute numerical error for 16-point chirp contours, shown as a function of the polar angle of W, discretized at $360^\circ/2048$. Each point represents the average error for ten random input vectors, each of unit length. The three red points indicate the errors for the three chirp contours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

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 estimated[18] that the inverse chirp transform can be performed in $2\mathsf{M}(n) + O(n)$, where $\mathsf{M}(n)$ 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 circle[19]. This problem is solved with approximate or iterative algorithms[19–22]. The accuracy and speed of approximate algorithms depend on the desired precision[20], which is often controlled by an oversampling parameter[22]. 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 defined[7] as follows:

$$\mathrm{X}_k = \sum_{j=0}^{N-1} \mathrm{x}_j \, A^{-j} \, W^{jk}, \qquad k = 0, 1, \ldots, M-1, \tag{1}$$

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:

$$\mathbf{X} = \boldsymbol{W}\,\mathbf{A}\,\mathbf{x}. \tag{2}$$

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

$$\mathbf{A} = \mathrm{diag}\left(A^{-0}, A^{-1}, \ldots, A^{-(N-1)}\right) \tag{3}$$

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

$$\boldsymbol{W} = \begin{bmatrix} W^{0\cdot 0} & W^{1\cdot 0} & \ldots & W^{(N-1)\cdot 0} \\ W^{0\cdot 1} & W^{1\cdot 1} & \ldots & W^{(N-1)\cdot 1} \\ \vdots & \vdots & \ddots & \vdots \\ W^{0\cdot (M-1)} & W^{1\cdot (M-1)} & \ldots & W^{(N-1)\cdot (M-1)} \end{bmatrix}. \tag{4}$$

Using Bluestein's substitution[9], i.e., $jk = \left(j^2 + k^2 - (k-j)^2\right)/2$, the matrix $\boldsymbol{W}$ can be expressed as $\boldsymbol{W} = \mathbf{P}\hat{\boldsymbol{W}}\mathbf{Q}$. That is, it is equal to a product of three matrices where $\hat{\boldsymbol{W}}$ is the following Toeplitz matrix

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

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

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

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

$$\mathbf{X} = \mathbf{P}\left(\hat{\boldsymbol{W}}\left(\mathbf{Q}\left(\mathbf{A}\,\mathbf{x}\right)\right)\right). \tag{7}$$

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 stated[10] by inverting Eq. (7), which leads to:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{Q}^{-1}\,\hat{\boldsymbol{W}}^{-1}\,\mathbf{P}^{-1}\,\mathbf{X}. \tag{8}$$

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 $\hat{\boldsymbol{W}}$ can be inverted[10] using a special case of the Gohberg–Semencul formula[23,24]. In other words, the inverse matrix $\hat{\boldsymbol{W}}^{-1}$ is given by:

$$\hat{\boldsymbol{W}}^{-1} = \frac{1}{\mathrm{u}_0}\left(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{A}}^T - \boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{D}}\right), \tag{9}$$

where $\boldsymbol{\mathcal{A}}$ is a lower-triangular Toeplitz matrix and $\boldsymbol{\mathcal{D}}$ is an upper-triangular Toeplitz matrix. Both $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{D}}$ are defined by the same generating vector $\mathbf{u} = (\mathrm{u}_0, \mathrm{u}_1, \mathrm{u}_2, \ldots, \mathrm{u}_{n-1})$, i.e.,

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} \mathrm{u}_0 & 0 & 0 & \ldots & 0 \\ \mathrm{u}_1 & \mathrm{u}_0 & 0 & \ldots & 0 \\ \mathrm{u}_2 & \mathrm{u}_1 & \mathrm{u}_0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{u}_{n-1} & \mathrm{u}_{n-2} & \mathrm{u}_{n-3} & \ldots & \mathrm{u}_0 \end{bmatrix}, \quad \boldsymbol{\mathcal{D}} = \begin{bmatrix} 0 & \mathrm{u}_{n-1} & \mathrm{u}_{n-2} & \ldots & \mathrm{u}_1 \\ 0 & 0 & \mathrm{u}_{n-1} & \ldots & \mathrm{u}_2 \\ 0 & 0 & 0 & \ldots & \mathrm{u}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix}. \tag{10}$$

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

$$\mathrm{u}_k = (-1)^k \frac{W^{\frac{2k^2-(2n-1)k+n(n-1)}{2}}}{\prod_{s=1}^{n-k-1}(W^s-1)\prod_{s=1}^{k}(W^s-1)}. \tag{11}$$

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

$$\mathbf{x} = \frac{1}{\mathrm{u}_0}\mathbf{A}^{-1}\mathbf{Q}^{-1}\left(\boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{A}}^T - \boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{D}}\right)\mathbf{P}^{-1}\mathbf{X}. \tag{12}$$

Algorithm 1 computes this expression in $O(n \log n)$ time by exploiting the structure of the matrices. See reference 10 for more details, proofs, and the pseudo-code for TOEPLITZMULTIPLYE and all of its dependencies.

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

ICZT($\mathbf{X}$, N, W, A)
$M \leftarrow$ LENGTH($\mathbf{X}$);
if $M \neq N$ **then**
  ERROR("M must be equal to N.");
end if
$n \leftarrow N$;
$\mathbf{x} \leftarrow$ EMPTYARRAY(n);
for $k \leftarrow 0$ **to** $n-1$ **do**
  $\mathrm{x}[k] \leftarrow W^{-\frac{k^2}{2}} \cdot \mathrm{X}[k]$; // multiply $\mathbf{P}^{-1}$ and $\mathbf{X}$
end for
// Precompute the necessary polynomial products.
$\mathbf{p} \leftarrow$ EMPTYARRAY(n);
$\mathrm{p}[0] \leftarrow 1$;
for $k \leftarrow 1$ **to** $n-1$ **do**
  $\mathrm{p}[k] \leftarrow \mathrm{p}[k-1] \cdot (W^k - 1)$;
end for
// Compute the generating vector $\mathbf{u}$.
$\mathbf{u} \leftarrow$ EMPTYARRAY(n);
for $k \leftarrow 0$ **to** $n-1$ **do**
  $\mathrm{u}[k] \leftarrow (-1)^k \dfrac{W^{\frac{2k^2-(2n-1)k+n(n-1)}{2}}}{\mathrm{p}[n-k-1] \cdot \mathrm{p}[k]}$;
end for
$\mathbf{z} \leftarrow$ ZEROVECTOR(n); // vector with n zeros
$\widehat{\mathbf{u}} \leftarrow (0, \mathrm{u}[n-1], \mathrm{u}[n-2], \ldots, \mathrm{u}[2], \mathrm{u}[1])$;
$\widetilde{\mathbf{u}} \leftarrow (\mathrm{u}[0], \underbrace{0, 0, \ldots, 0}_{n-1 \text{ zeros}})$;
$\mathbf{x}' \leftarrow$ TOEPLITZMULTIPLYE $(\widehat{\mathbf{u}}, \mathbf{z}, \mathbf{x})$; // $\boldsymbol{\mathcal{D}}$
$\mathbf{x}' \leftarrow$ TOEPLITZMULTIPLYE $(\mathbf{z}, \widehat{\mathbf{u}}, \mathbf{x}')$; // $\boldsymbol{\mathcal{D}}^T$
$\mathbf{x}'' \leftarrow$ TOEPLITZMULTIPLYE $(\mathbf{u}, \widetilde{\mathbf{u}}, \mathbf{x})$; // $\boldsymbol{\mathcal{A}}^T$
$\mathbf{x}'' \leftarrow$ TOEPLITZMULTIPLYE $(\widetilde{\mathbf{u}}, \mathbf{u}, \mathbf{x}'')$; // $\boldsymbol{\mathcal{A}}$
for $k \leftarrow 0$ **to** $n-1$ **do**
  $\mathrm{x}[k] \leftarrow \dfrac{\mathrm{x}''[k] - \mathrm{x}'[k]}{\mathrm{u}[0]}$; // subtract and divide by u_0
end for
for $k \leftarrow 0$ **to** $n-1$ **do**
  $\mathrm{x}[k] \leftarrow A^k \cdot W^{-\frac{k^2}{2}} \cdot \mathrm{x}[k]$; // multiply by $\mathbf{A}^{-1}\mathbf{Q}^{-1}$
end for
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 $F_1, F_2, \ldots, F_{n-1}$. This property becomes more clear if the Farey sequences shown above are rewritten with extra space between some of the terms as shown below:

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

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 $F_1, F_2, \ldots, F_{n-1}$ 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 $\mathrm{e}^{i2\pi p/q}$ 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 $q \in \{1, 2, \ldots, n\}$ such that the complex exponential $\mathrm{e}^{i\theta}$ 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 $\mathrm{e}^{i\theta}$ is a root of unity of order q. That is,

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

because p is an integer.

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

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

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,

$$\theta = \frac{2\pi p}{q}, \tag{15}$$

which implies that $\theta \in F_q \subseteq F_n$, as required. □

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 chirp 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 $W = \mathrm{e}^{i2\pi \frac{p}{q}}$. 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 $k \in \{0, 1, 2, \ldots, n-1\}$.

($\Rightarrow$) 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,

$$\begin{aligned}\prod_{s=1}^{n-1}(W^s - 1) &= \prod_{s=1,\, s\neq q}^{n-1}(W^s - 1)\prod_{s=q}^{q}(W^s - 1) \\ &= \prod_{s=1,\, s\neq q}^{n-1}\left(\mathrm{e}^{i2\pi \frac{ps}{q}} - 1\right)\underbrace{\left(\mathrm{e}^{i2\pi \frac{pq}{q}} - 1\right)}_{0} = 0.\end{aligned} \tag{16}$$

($\Leftarrow$) 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 $s \in \{1, 2, \ldots, n-1\}$. Therefore, each term in the denominator of each element u_k is non-zero. □

Supplementary Section S4 gives additional proofs and examples that perform the analysis using the rank of the Vandermonde matrix $\boldsymbol{W}$ 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 $\boldsymbol{W}$ for chirp 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 $\boldsymbol{W}$. This was done using the following formula: $\mathrm{cond}(\boldsymbol{W}) = \sigma_{\max}/\sigma_{\min}$, where $\sigma_{\max}$ is the largest singular value and $\sigma_{\min}$ is the smallest singular value of the matrix $\boldsymbol{W}$.

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 chirp 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 chirp 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 chirp contours are defined in terms of the z-transform, which uses negative powers (see Supplementary Section S1).

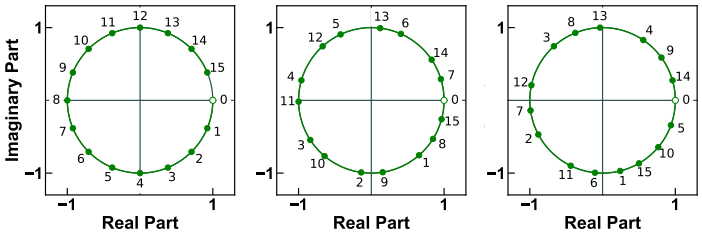

Figure 3. Three chirp contours with 16 points on the unit circle. The contour points are numbered in increasing order to illustrate the cases when a contour completes more than one revolution over the unit circle. From left to right, the polar angle of the parameter W is: 22.5°, 49.0°, and 76.0°.

Figure 4a shows the absolute numerical error of the CZT–ICZT procedure (see Methods) for 16-point contours on the unit circle. Figure 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 $\{\frac{360^\circ}{15}, \frac{360^\circ}{14}, \frac{360^\circ}{13}, \ldots, \frac{360^\circ}{8}\}$. These angles form a subset of the harmonic sequence $(\frac{360^\circ}{1}, \frac{360^\circ}{2}, \frac{360^\circ}{3}, \ldots)$, 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 numbers[25] approximate these angles and their complex exponentials. Because π is 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 $\boldsymbol{W}$ 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

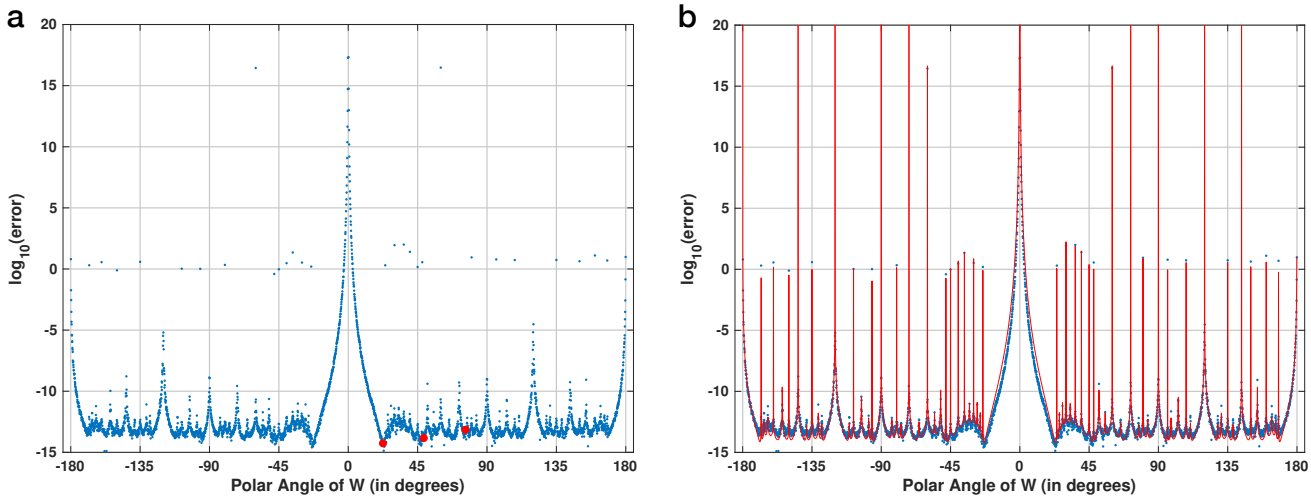

Figure 4. (**a**) Absolute numerical error of the CZT–ICZT procedure, as a function of the polar angle of W for 16-point chirp contours on the unit circle. The discretization step was 0.1°, i.e., there were 3600 angles. The three red points indicate the error for the three contours shown in Fig. 3. (**b**)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

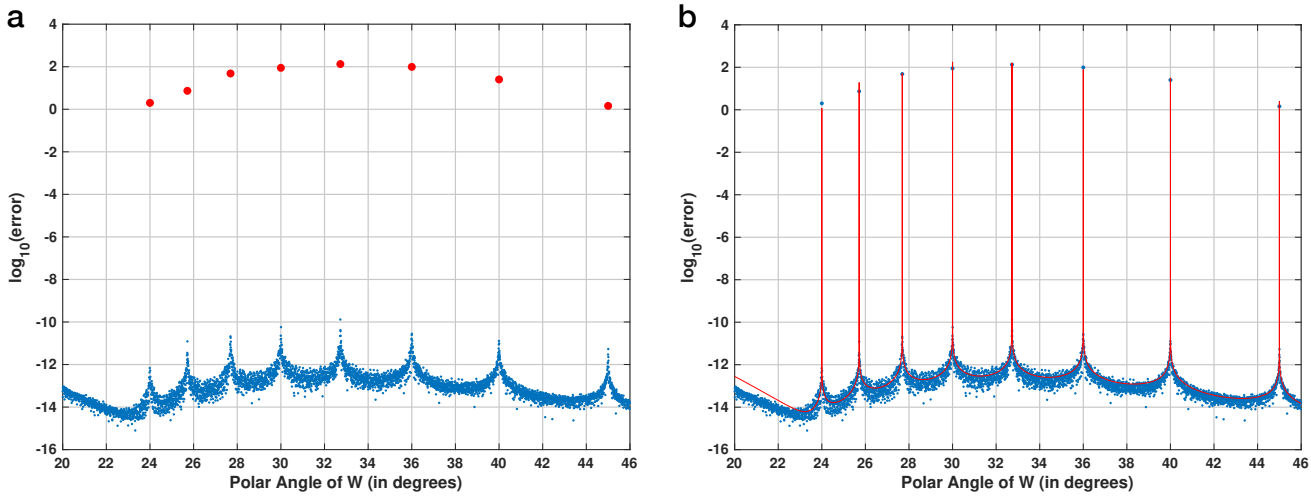

Figure 5. (**a**) Close-up of the harmonic hedgehog located between the first two red points in Fig. 4a. The numerical error spikes when the polar angle of W is close to an element of the set $\{\frac{360^\circ}{15}, \frac{360^\circ}{14}, \frac{360^\circ}{13}, \ldots, \frac{360^\circ}{8}\}$. The eight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

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.

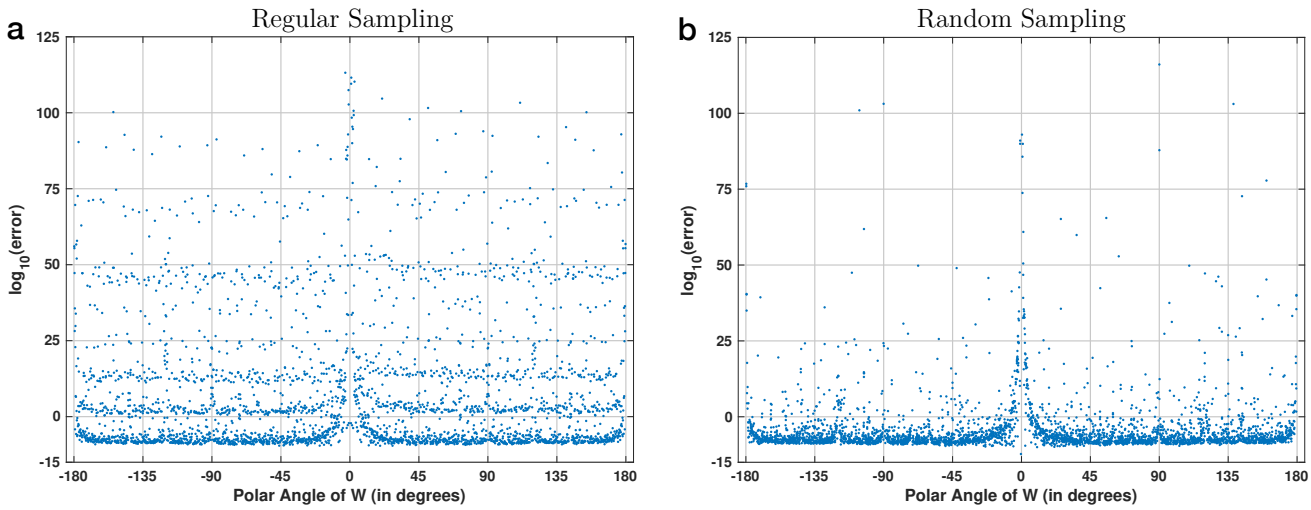

Figure 6. Absolute numerical error of the CZT–ICZT procedure for chirp contours with 1024 points on the unit circle. The results in (**a**) are for 3600 regularly-spaced angles with a discretization step of 0.1°. This discretization hits many Farey angles of lower orders, which leads to the layering 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**).

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 Figs. 7g and 7h. 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 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 increases[26]. 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 reference 27, p. 268 and reference 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 Figs. 7c, 7f, and 7i don't hit any small-order Farey angles exactly. This is true for any regular discretization in 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.

n	16	32	64	128	256	512	1024	2048
$\lvert F_{n-1}\rvert$	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 $|W| = 1$.

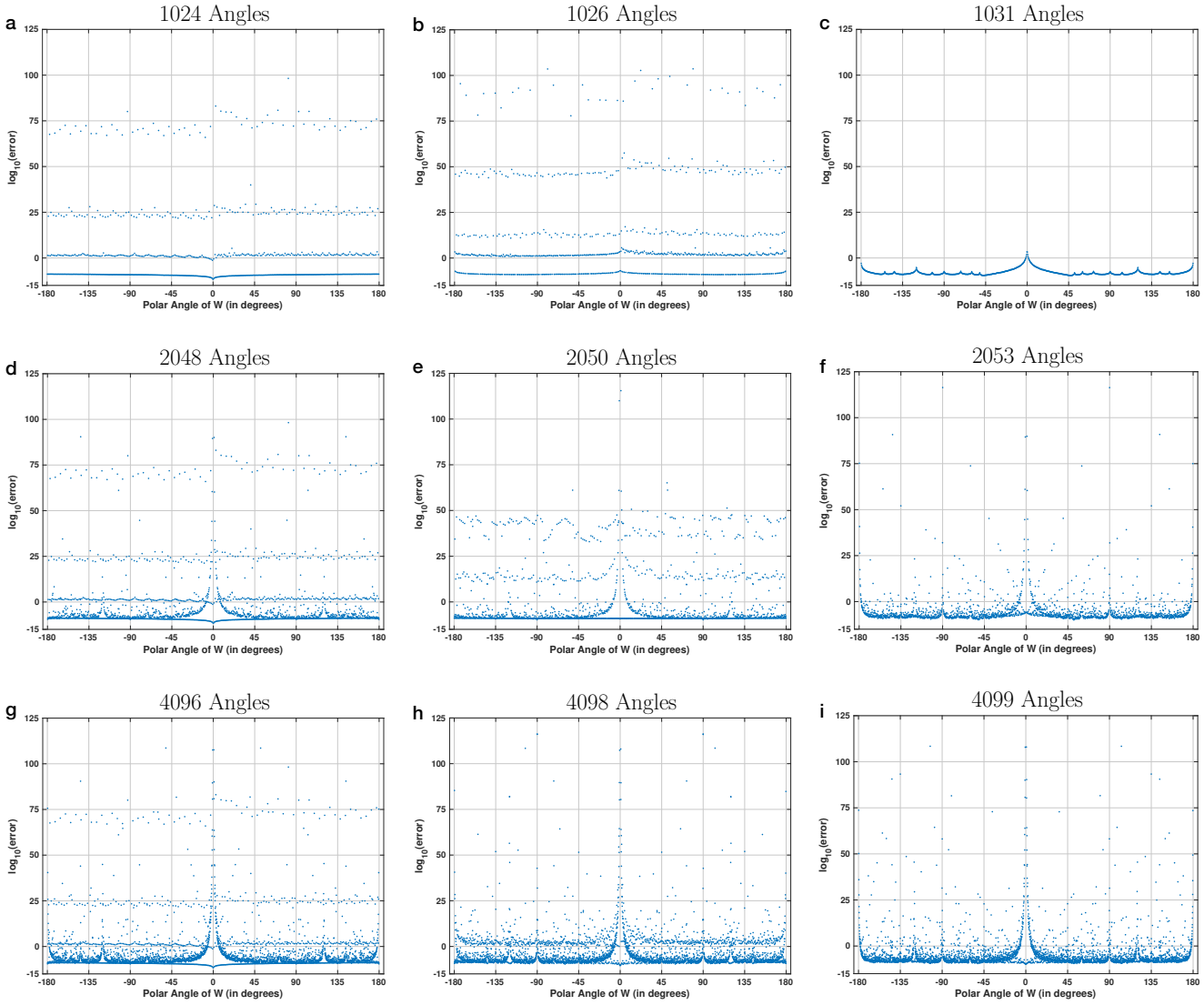

Figure 7. Absolute numerical error as a function of the polar angle of the transform parameter W, plotted for chirp contours with 1024 points on the unit circle. The nine plots illustrate the variety of shapes that the error function can take depending on the choice of discretization. In all plots, the 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**), (**b**), and (**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**), and (**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.

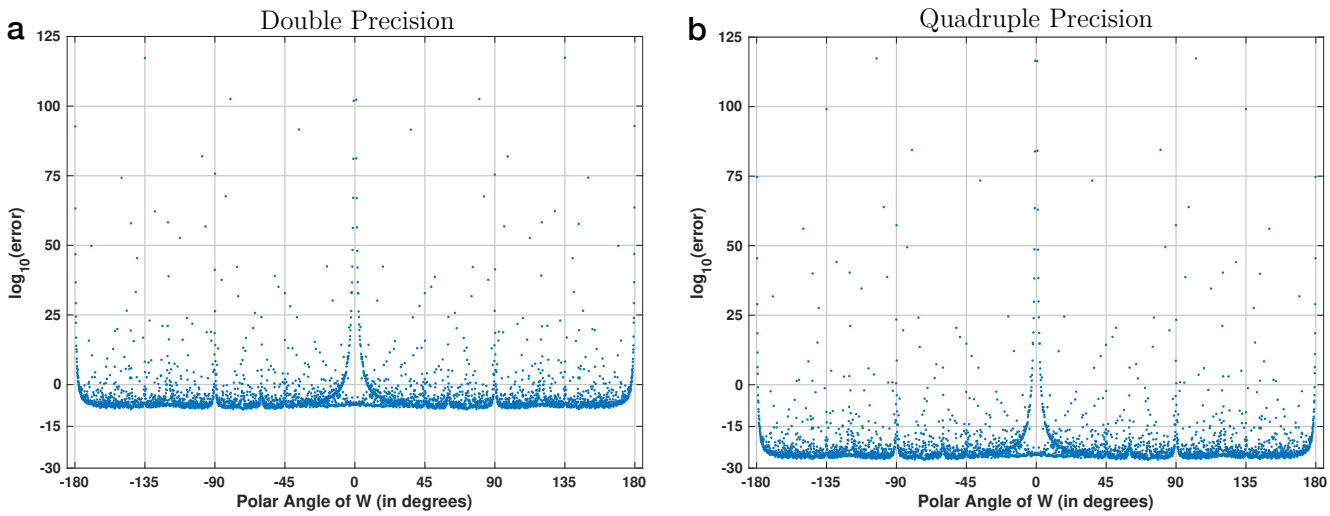

Figure 8. Absolute numerical error for the CZT–ICZT procedure, computed with two different floating-point precisions. In both plots, the results are for 2048-point contours on the unit circle and for 4099 polar angles of W that were discretized using regular intervals. The results in (**a**) were computed 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* library[33] through an interface in *Boost Multiprecision*[34].

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 points[10]. This is consistent with previous observations that Vandermonde systems can be ill-conditioned and should be solved with double precision or higher[29]. 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. Fig. 8a shows the results for double precision, which is implemented natively in modern CPUs. Fig. 8b shows a different plot that was computed with quadruple precision. It was generated using the *libquadmath* library[33] 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 reference 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:

$$U_1 = \log \sqrt{\sum_{k=1}^{N-1} |\mathrm{u}_k|^2} = \frac{1}{2} \log \sum_{k=1}^{N-1} |\mathrm{u}_k|^2, \qquad U_2 = \log \sqrt{\sum_{k=0}^{N-1} |\mathrm{u}_k|^2} = \frac{1}{2} \log \sum_{k=0}^{N-1} |\mathrm{u}_k|^2, \qquad U_3 = -\log |\mathrm{u}_0|. \quad (17)$$

The term U_1 is equal to the logarithm of the Euclidean norm of the first row of the matrix $\boldsymbol{\mathcal{D}}$, 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 column of the matrix $\boldsymbol{\mathcal{A}}$,

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 $\frac{1}{|\mathrm{u}_0|}$, 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 terms[10]. 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:

$$T = \frac{3}{2}\log N. \tag{18}$$

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:

$$B = -p\log 2 + C_1 \log N + C_2, \tag{19}$$

where C_1 and C_2 are implementation-dependent constants. For 64-bit and 128-bit IEEE-754 floating-point numbers[25], 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,

$$E = \|\hat{\mathbf{x}} - \mathbf{x}\|. \tag{20}$$

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

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

Error formula for the ICZT followed by the CZT: The absolute error of the ICZT–CZT procedure is

$$E = \|\hat{\mathbf{X}} - \mathbf{X}\|, \tag{22}$$

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:

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

Evaluation of the Error Prediction Formulas: The accuracies of Eq. (21) and Eq. (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.

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.

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 reference 13 that is reproduced below:

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

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:

$$\mathbf{y} = \overline{\boldsymbol{W}\,\mathbf{A}\,\overline{\mathbf{X}}} = \overline{\boldsymbol{W}}\;\overline{\mathbf{A}}\;\mathbf{X}. \tag{25}$$

In this formula, $\mathbf{y}$ is the resulting vector, $\overline{\boldsymbol{W}}$ is the matrix obtained by conjugating all elements of the Vandermonde matrix $\boldsymbol{W}$ 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 A and W. That is, the vector $\mathbf{y}$ in Eq. (25) can be expressed as:

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

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.

Reference 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.,

$$\hat{\mathbf{x}} = \frac{\mathbf{y}}{\|\mathbf{y}\|}. \tag{27}$$

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 $\log_{10}\|\mathbf{x} - \hat{\mathbf{x}}\| \le \log_{10} 2 \approx 0.3$, 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.

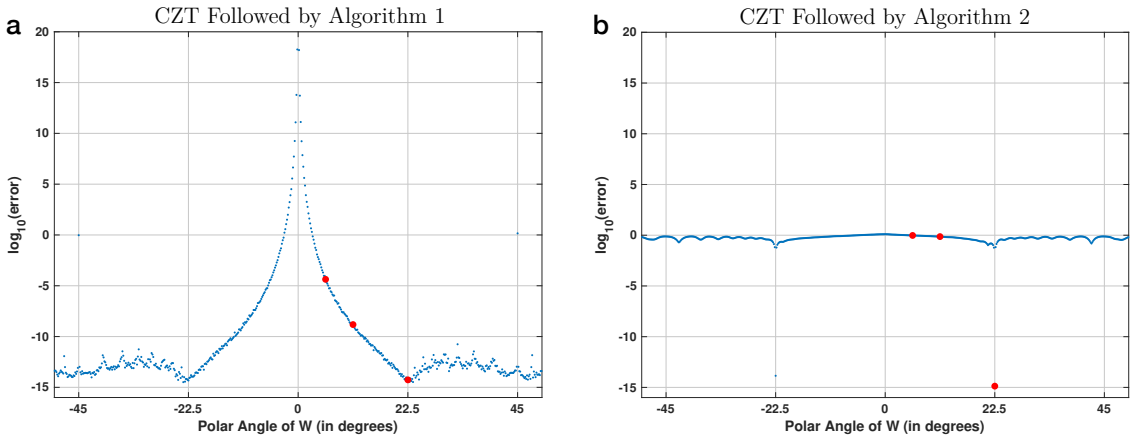

Figure 9. Absolute numerical error for 16-point chirp contours, shown as a function of the polar angle of W for two different experimental procedures: (**a**) the CZT followed by Algorithm 1, which implements the ICZT (this is the same as Fig. 2); and (**b**) the CZT followed by Algorithm 2, which uses the contour reversal approach proposed in reference 13. 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.

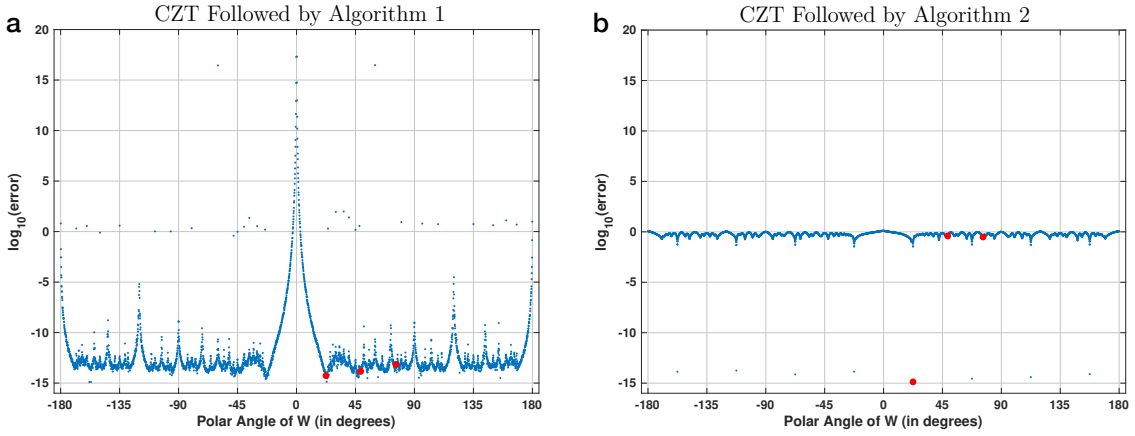

Figure 10. Visualization of the accuracy of two experimental procedures for 16-point chirp contours. This is similar to Fig. 9, but in this case the discretization step for the polar angle of W was set to 0.1°. The three red points in both plots correspond to the chirp contours shown in Fig. 3. The plot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

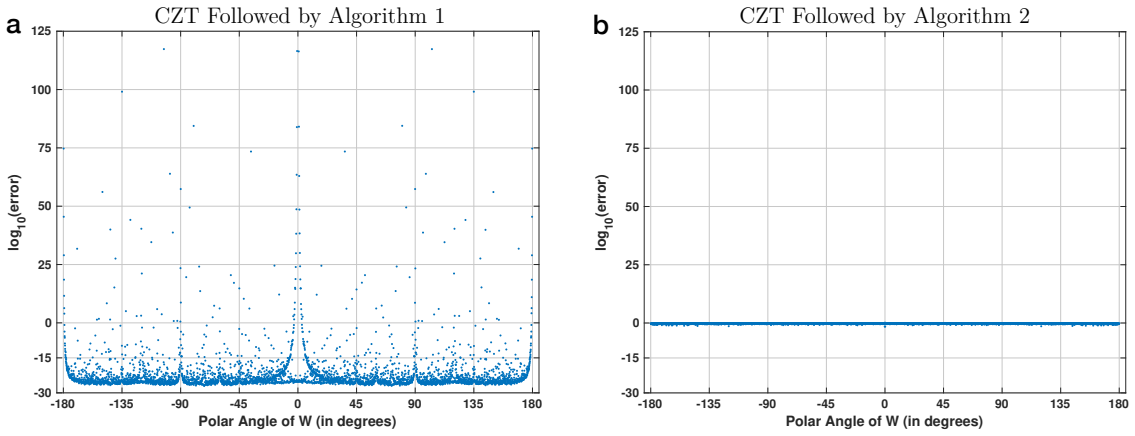

Figure 11. Visualization of the accuracy of two experimental procedures for 2048-point chirp contours. This is similar to Fig. 10, but in this case the discretization used 4099 regularly-spaced polar angles of W and the results were computed using quadruple precision (i.e., 128 bits) instead of double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

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 paper[10]. 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 vector[10], 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* library[34]. In both cases the floating-point numbers were stored in IEEE-754 format[25].

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 $[0°, 360°)$. 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 $[180°, 360°)$ is plotted as negative angles in the interval $[-180°, 0°)$.

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

$$\theta_k = {}^{2\pi k}/_{P}, \quad \text{where } k \in \{0, 1, 2, \ldots, P-1\}. \tag{28}$$

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: $W = \mathrm{e}^{i\theta} = \cos\theta + i \sin\theta$.

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(X, N, W, A)
2: y ← CZT(X, N, W̄, Ā);    // reverse the chirp contour
3: d ← 0;
4: for k ← 0 to N−1 do
5:     d ← d + |y[k]|²;
6: end for
7: x̂ ← EMPTYVECTOR(N);
8: for k ← 0 to N−1 do
9:     x̂[k] ← y[k]/√d;
10: end for
11: return x̂;
```

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 algorithm[38]. Algorithm 3 shows the pseudo-code for an efficient procedure[38] 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: F ← EMPTYLIST();
3: (a, b, c, d) ← (0, 1, 1, n);
4: F ← APPEND(F, a/b);
5: while c ≤ n do
6:     k ← ⌊(n+b)/d⌋;
7:     (a, b, c, d) ← (c, d, kc − a, kd − b);
8:     F ← APPEND(F, a/b);
9: end while
10: return 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 $\mathbf{a} = (\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_{m-1})$ and $\mathbf{b} = (\mathrm{b}_0, \mathrm{b}_1, \mathrm{b}_2, \ldots, \mathrm{b}_{m-1})$, the R^2 value was computed as follows:

$$R^2 = 1 - \alpha/\beta, \quad \text{where} \quad \alpha = \sum_{k=0}^{m-1} \Big(\big(\mathrm{a}_k - \overline{a}\,\big) - \big(\mathrm{b}_k - \overline{b}\,\big) \Big)^2 \quad \text{and} \quad \beta = \sum_{k=0}^{m-1} \big(\mathrm{b}_k - \overline{b}\,\big)^2. \tag{29}$$

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

$$\overline{a} = \big(\mathrm{a}_0 + \mathrm{a}_1 + \mathrm{a}_2 + \cdots + \mathrm{a}_{m-1}\big)/m \quad \text{and} \quad \overline{b} = \big(\mathrm{b}_0 + \mathrm{b}_1 + \mathrm{b}_2 + \cdots + \mathrm{b}_{m-1}\big)/m. \tag{30}$$

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 $\overline{b} - \overline{a}$ 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 substitution[9] 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 vector[10]. 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 vector[10]. 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 reference 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 Pustylnikov's decomposition[40,41], see also reference 42, p. 40 and reference 43, p. 66.

Both subroutines multiply a circulant matrix or a skew-circulant matrix[44] 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.

References

1. Rockmore, D. The FFT: an algorithm the whole family can use. *Comput. Sci. & Eng.* **2**, 60–64 (2000).
2. Dongarra, J. & Sullivan, F. Introduction to the top 10 algorithms. *Comput. Sci. & Eng.* **2**, 22–23 (2000).
3. Cooley, J. & Tukey, J. An algorithm for the machine calculation of complex Fourier series. *Math. 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 Syst. Tech. J.* **48**, 1249–1292 (1969).
7. Rabiner, L., Schafer, R. & Rader, C. The chirp z-transform algorithm. *IEEE Transactions on Audio Electroacoustics* **17**, 86–92 (1969).
8. Rabiner, L. The chirp z-transform – a lesson in serendipity. *IEEE Signal Process. Mag.* **21**, 118–119 (2004).
9. Bluestein, L. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Transactions on Audio Electroacoustics* **18**, 451–455 (1970).
10. Sukhoy, V. & Stoytchev, A. Generalizing the inverse FFT off the unit circle. *Sci. Reports* **9**, 14443 (2019).
11. Mersereau, R. *Digital reconstruction of multidimensional signals from their projections*. Ph.D. thesis, Massachusetts Institute of Technology (1973). Chapter 5.
12. Mersereau, R. An algorithm for performing an inverse chirp z-transform. *IEEE Transactions on Acoust. Speech Signal Process.* **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 Tomsk State Univ. Control. Syst. Radioelectron.* **24**, 162–165, Part 1 (2011).
15. Oppenheim, A. & Schafer, R. *Discrete-Time Signal Processing* (Prentice Hall, Upper Saddle River, NJ, 2009), 3 edn. Chapter 9, pp. 749–750.
16. Bailey, D. & Swarztrauber, P. The fractional Fourier transform and applications. *SIAM Rev.* **33**, 389–404 (1991).
17. Swarztrauber, P., Sweet, R., Briggs, W. & Otto, J. Bluestein's FFT for arbitrary N on the hypercube. *Parallel Comput.* **17**, 607–617 (1991).
18. Bostan, A. & Schost, É. Polynomial evaluation and interpolation on special sets of points. *J. Complex.* **21**, 420–446 (2005).
19. Dutt, A. & Rokhlin, V. Fast Fourier transforms for nonequispaced data. *SIAM J. on Sci. Comput.* **14**, 1368–1393 (1993).
20. Dutt, A. & Rokhlin, V. Fast Fourier transforms for nonequispaced data, II. *Appl. Comput. Harmon. Analysis* **2**, 85–100 (1995).
21. Kircheis, M. & Potts, D. Direct inversion of the nonequispaced fast Fourier transform. *Linear Algebr. its Appl.* **575**, 106–140 (2019).
22. Ruiz-Antolín, D. & Townsend, A. A nonuniform fast Fourier transform based on low rank approximation. *SIAM J. on Sci. Comput.* **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 (American Mathematical Society, Providence, RI, 1974). Section III.6.
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 (2019). https://oeis.org/A005728.
27. Hardy, G. & Wright, E. *An Introduction to the Theory of Numbers* (Oxford University Press, London, 1975), 4 edn.
28. Conway, J. & Guy, R. *The Book of Numbers* (Copernicus, an imprint of Springer-Verlag, Inc., New York, 1996), corrected edn. pp. 152–156.
29. Press, W., Teukolsky, S., Vetterling, W. & Flannery, B. *Numerical Recipes* (Cambridge Univ. Press, Cambridge, MA, 2007), 3 edn. p. 95.
30. Björck, Å. & Pereyra, V. Solution of Vandermonde systems of equations. *Math. Comput.* **24**, 893–903 (1970).
31. Higham, N. Error analysis of the Björck–Pereyra algorithms for solving Vandermonde systems. *Numer. Math.* **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 Appl.* **27**, 142–152 (2005).
33. Free Software Foundation. *The GCC Quad-Precision Math Library* (2018). https://gcc.gnu.org/onlinedocs/gcc-8.2.0/libquadmath.pdf.
34. Maddock, J. & Kormanyos, C. *Boost Multiprecision* (2018). https://www.boost.org/doc/libs/1_69_0/libs/multiprecision/doc/html/.
35. Farey, J. On a curious property of vulgar fractions. *The Philos. Mag.* **47**, 385–386 (1816).
36. Niven, I. & Zuckerman, H. *An Introduction to the Theory of Numbers* (Wiley, New York, 1980), 4 edn. Chapter 6: Farey Fractions and Irrational Numbers, pp. 171–185.
37. Graham, R., Knuth, D. & Patashnik, O. *Concrete Mathematics* (Addison-Wesley, Reading, MA, 1994), 2 edn.
38. Routledge, N. Computing Farey series. *The Mathematical Gazette* **92**, 55–62 (2008).
39. Golub, G. & Van Loan, C. *Matrix Computations* (Johns Hopkins University Press, Baltimore, MD, 1996), 3 edn.
40. Pustylnikov, L. On the algebraic structure of the spaces of Toeplitz and Hankel matrices (in Russian). *Sov. Math. Dokl.* **21**, 141–144 (1980).
41. Pustylnikov, 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).

Acknowledgments

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.

Additional Information

Supplementary information was submitted together with 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.

Submitted: November 3, 2019; Accepted: February 9, 2020.

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

Vladimir Sukhoy[1] & Alexander Stoytchev[1,*]

[1] Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA. Correspondence and requests for materials should be addressed to A.S. (email: alexs@iastate.edu).

S1. Definitions for Contours and Transforms

By mathematical convention, positive angles correspond to counter-clockwise rotations and negative angles correspond to clockwise rotations (i.e., the right-hand rule). This convention, however, is violated for the winding direction of chirp contours. That is, *positive polar angles of W correspond to clockwise rotations*. The reason for this break with convention is that the CZT was defined with the z-transform, which uses negative powers, instead of the power series, which uses positive powers. The right-hand rule, however, still holds for the polar angle of the transform parameter A. This section clarifies these technical details using examples and explicit definitions.

A contour is a list of complex numbers that is derived from the transform parameters. Each complex number specifies a frequency component vector, the elements of which are equal to the integer powers of this number. For the DFT and the CZT, each element of the output vector is equal to the complex inner product between the input vector and the corresponding frequency component vector. The inverse transforms, i.e., the IDFT and the ICZT, map the output vector back to the input vector. A contour is shared by a matching pair of forward and inverse transforms, i.e., when they have the same parameters.

We will start by defining the Fourier contour that is used by the DFT and the IDFT (and, indirectly, by their fast implementations using the FFT and the IFFT algorithms).

Definition 3. ***Fourier contour.***
A Fourier contour consists of N points $\mathrm{c}_0, \mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_{N-1}$ in the complex plane, arranged in a counter-clockwise direction on the unit circle starting from the point $(1, 0)$. The coordinates of the k-th point, c_k, are given by the following formula:

$$\mathrm{c}_k = \mathrm{e}^{\frac{i2\pi k}{N}} = \cos\left(\frac{2\pi k}{N}\right) + i \sin\left(\frac{2\pi k}{N}\right), \qquad (31)$$

for each $k \in \{0, 1, 2, \ldots, N-1\}$.

For a DFT of size N, the Fourier contour points are equal to the N complex roots of unity of order N. The ordering of the contour points matches the way in which the roots are enumerated. Also, the angle between two consecutive points is positive, which is in agreement with the right-hand rule.

The complex inner product between two complex vectors $\mathbf{a}$ and $\mathbf{b}$ of length N is defined as follows:

$$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{n=0}^{N-1} \mathrm{a}_n \cdot \overline{\mathrm{b}_n}\,. \qquad (32)$$

Notice that the elements of the second vector are conjugated. Thus, changing the order of the two vectors in the inner product is equivalent to conjugating its value, i.e.,

$$\langle \mathbf{b}, \mathbf{a} \rangle = \overline{\langle \mathbf{a}, \mathbf{b} \rangle}. \qquad (33)$$

This definition ensures that the complex inner product of a vector with itself is always nonnegative and is zero if and only if the vector is zero.

The DFT output vector $\mathbf{X}$ contains the values of N complex inner products between the DFT input vector $\mathbf{x}$ and each of the frequency component vectors $\mathbf{v}^{(0)}, \mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \ldots, \mathbf{v}^{(N-1)}$. Each vector $\mathbf{v}^{(k)}$ is defined as follows:

$$\mathbf{v}^{(k)} = \left(\left(\mathrm{e}^{\frac{i2\pi k}{N}}\right)^0, \left(\mathrm{e}^{\frac{i2\pi k}{N}}\right)^1, \ldots, \left(\mathrm{e}^{\frac{i2\pi k}{N}}\right)^{N-1}\right). \qquad (34)$$

Definition 4. ***Discrete Fourier Transform (DFT).***
Let $\mathbf{c} = (\mathrm{c}_0, \mathrm{c}_1, \mathrm{c}_2, \ldots, \mathrm{c}_{N-1})$ be the Fourier contour for a transform of size N. Let $\mathbf{x} = (\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{N-1})$ be a complex input vector of length N. The DFT of the vector $\mathbf{x}$ is a complex vector $\mathbf{X}$, also of length N. For each $k \in \{0, 1, 2, \ldots, N-1\}$, the value of X_k is equal to the complex inner product between the vector $\mathbf{x}$ and the k-th frequency component vector $\mathbf{v}^{(k)}$. That is,

$$\mathrm{X}_k = \langle \mathbf{x}, \mathbf{v}^{(k)} \rangle = \sum_{n=0}^{N-1} \mathrm{x}_n \overline{\left(\mathrm{e}^{\frac{i2\pi k}{N}}\right)^n} = \sum_{n=0}^{N-1} \mathrm{x}_n \, \mathrm{e}^{-\frac{i2\pi kn}{N}}. \qquad (35)$$

Figure S1 shows two different contours that can be derived from Eq. (35). Only the blue contour in Fig. S1a is consistent with Definition 3. The red contour in Fig. S1b is derived from the blue contour by conjugating all of its sampling points. This contour traverses the unit circle in the clockwise direction. The following color version of Eq. (35) visualizes the link between these two contours:

$$\mathrm{X}_k = \sum_{n=0}^{N-1} \mathrm{x}_n \overline{\left(\mathrm{e}^{\frac{i2\pi k}{N}}\right)^n} = \sum_{n=0}^{N-1} \mathrm{x}_n \left(\mathrm{e}^{-\frac{i2\pi k}{N}}\right)^n. \qquad (36)$$

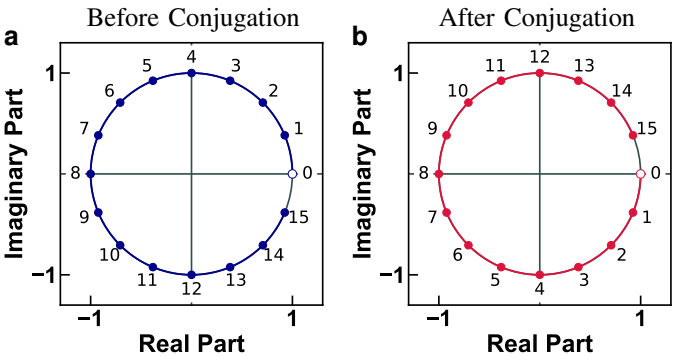

Fig. S1. Two 16-point contours based on two different interpretations of Eq. (35): (**a**) before the conjugation has been applied; and (**b**) after the conjugation has been applied. Only the contour in (**a**) is consistent with Definition 3.

Most references don't mention the conjugation and define the DFT using the rightmost sum in Eq. (35). Even though this is technically correct, it obscures the link to the Fourier contour and can lead to a lot of confusion. For example, without the conjugation it is more difficult to explain the essence of positive and negative frequencies and how the DFT/FFT orders them. Definition 4 makes things more clear by formulating the transform in terms of the complex inner product.

The IDFT inverts the DFT for the same contour. Definition 5 gives the formula for this inverse transform.

Definition 5. ***Inverse DFT (IDFT).***
The IDFT of a complex vector $\mathbf{X}$ of length N is a complex vector $\mathbf{x}$, also of length N. Its n-th element is given by:

$$\mathrm{x}_n = \frac{1}{N} \sum_{k=0}^{N-1} \mathrm{X}_k \, \mathrm{e}^{\frac{i2\pi kn}{N}}. \qquad (37)$$

The following example illustrates the DFT of size 4. The elements of the output vector $\mathbf{X} = (\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)$ are defined with the following four complex inner products:

$$\begin{aligned} \mathrm{X}_0 &= \langle \mathbf{x}, \mathbf{v}^{(0)} \rangle, \\ \mathrm{X}_1 &= \langle \mathbf{x}, \mathbf{v}^{(1)} \rangle, \\ \mathrm{X}_2 &= \langle \mathbf{x}, \mathbf{v}^{(2)} \rangle, \\ \mathrm{X}_3 &= \langle \mathbf{x}, \mathbf{v}^{(3)} \rangle. \end{aligned} \tag{38}$$

Using matrix notation, the output vector $\mathbf{X}$ can be expressed as the matrix–vector product between the DFT matrix $\mathbf{F}$ and the input vector $\mathbf{x}$, i.e., $\mathbf{X} = \mathbf{F}\,\mathbf{x}$. That is,

$$\underbrace{\begin{bmatrix} \mathrm{X}_0 \\ \mathrm{X}_1 \\ \mathrm{X}_2 \\ \mathrm{X}_3 \end{bmatrix}}_{\mathbf{X}} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \omega & \omega^2 & \omega^3 \\ 1 & \omega^2 & \omega^4 & \omega^6 \\ 1 & \omega^3 & \omega^6 & \omega^9 \end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix} \mathrm{x}_0 \\ \mathrm{x}_1 \\ \mathrm{x}_2 \\ \mathrm{x}_3 \end{bmatrix}}_{\mathbf{x}}, \tag{39}$$

where $\omega = \overline{\mathrm{e}^{\frac{i2\pi}{N}}} = \mathrm{e}^{-\frac{i2\pi}{N}}$.

Each row of the matrix $\mathbf{F}$ can be obtained by conjugating all elements of the corresponding frequency component vector defined in Eq. (34). That is,

$$\mathbf{F} = \begin{bmatrix} \overline{\mathrm{v}_0^{(0)}} & \overline{\mathrm{v}_1^{(0)}} & \overline{\mathrm{v}_2^{(0)}} & \overline{\mathrm{v}_3^{(0)}} \\ \overline{\mathrm{v}_0^{(1)}} & \overline{\mathrm{v}_1^{(1)}} & \overline{\mathrm{v}_2^{(1)}} & \overline{\mathrm{v}_3^{(1)}} \\ \overline{\mathrm{v}_0^{(2)}} & \overline{\mathrm{v}_1^{(2)}} & \overline{\mathrm{v}_2^{(2)}} & \overline{\mathrm{v}_3^{(2)}} \\ \overline{\mathrm{v}_0^{(3)}} & \overline{\mathrm{v}_1^{(3)}} & \overline{\mathrm{v}_2^{(3)}} & \overline{\mathrm{v}_3^{(3)}} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{bmatrix}. \tag{40}$$

The CZT also uses a contour, but its points are no longer restricted to lie on the unit circle. Its shape and winding direction are defined by the parameters A, W, and M.

Definition 6. ***Chirp contour.***
Let A and W be two complex numbers and let M be a positive integer. A chirp contour consists of M complex points $\mathrm{z}_0, \mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_{M-1}$ that lie on a logarithmic spiral. The starting point, z_0, is equal to A. The remaining points are derived by multiplying the parameter A with the negative integer powers of the parameter W. That is,

$$\mathrm{z}_k = A\,W^{-k}, \qquad \textit{for each } k \in \{0, 1, 2, \ldots, M-1\}. \tag{41}$$

Each element X_k of the CZT output vector $\mathbf{X}$ is defined as the z-transform of the input vector $\mathbf{x}$, where the value of the z-transform is evaluated at the corresponding contour point z_k.

Definition 7. ***Chirp Z-Transform (CZT).***
Let $\mathbf{z} = (\mathrm{z}_0, \mathrm{z}_1, \mathrm{z}_2, \ldots, \mathrm{z}_{M-1})$ be the chirp contour. Let $\mathbf{x} = (\mathrm{x}_0, \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_{N-1})$ be the complex input vector of length N. Each element X_k of the CZT complex output vector $\mathbf{X} = (\mathrm{X}_0, \mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{M-1})$ is equal to the value of the z-transform of the input vector $\mathbf{x}$ at the corresponding contour point z_k. That is,

$$\mathrm{X}_k = \sum_{j=0}^{N-1} \mathrm{x}_j\, \mathrm{z}_k^{-j} = \sum_{j=0}^{N-1} \mathrm{x}_j (A\,W^{-k})^{-j} = \sum_{j=0}^{N-1} \mathrm{x}_j A^{-j}\, W^{jk}. \tag{42}$$

Similarly to the DFT, the CZT can also be defined using the complex inner product. In this case, the frequency component vectors are formed by conjugating the negative powers of the contour points. In other words, the CZT frequency components $\hat{\mathbf{v}}^{(0)}, \hat{\mathbf{v}}^{(1)}, \hat{\mathbf{v}}^{(2)}, \ldots, \hat{\mathbf{v}}^{(M-1)}$ can be expressed as follows:

$$\hat{\mathbf{v}}^{(k)} = \Big(\overline{\big(A\,W^{-k}\big)^{-0}}, \overline{\big(A\,W^{-k}\big)^{-1}}, \ldots, \overline{\big(A\,W^{-k}\big)^{-(N-1)}} \Big), \tag{43}$$

where $k \in \{0, 1, 2, \ldots, M-1\}$.

Each element X_k of the CZT output vector $\mathbf{X}$ can be expressed as the inner product between the CZT input vector $\mathbf{x}$ and the corresponding frequency component $\hat{\mathbf{v}}^{(k)}$. That is,

$$\begin{aligned} \mathrm{X}_k = \langle \mathbf{x}, \hat{\mathbf{v}}^{(k)} \rangle &= \sum_{j=0}^{N-1} \mathrm{x}_j\, \overline{\hat{\mathrm{v}}_j^{(k)}} = \sum_{j=0}^{N-1} \mathrm{x}_j\, \overline{\overline{\big(A\,W^{-k}\big)^{-j}}} \\ &= \sum_{j=0}^{N-1} \mathrm{x}_j\, (A\,W^{-k})^{-j}. \end{aligned} \tag{44}$$

For the case when the CZT is equivalent to the DFT, i.e., $A = 1$, $W = \mathrm{e}^{-\frac{i2\pi}{N}}$, and $M = N$, the frequency components of the CZT are equal to those of the DFT. That is,

$$\hat{\mathrm{v}}_j^{(k)} = \overline{\big(A\,W^{-k}\big)^{-j}} = \overline{\Big(1 \cdot \mathrm{e}^{\frac{i2\pi k}{N}}\Big)^{-j}} = \mathrm{e}^{\frac{i2\pi kj}{N}} = \mathrm{v}_j^{(k)}, \tag{45}$$

for each $k \in \{0, 1, \ldots, N-1\}$ and each $j \in \{0, 1, \ldots, N-1\}$. The chirp contour in this case matches the Fourier contour, e.g., for $M = 16$ it is equal to the blue contour in Fig. S1a.

Figure S2 gives an example with two chirp contours that lie on the unit circle. In both cases, the polar angle of A is positive and is equal to $30°$ and $45°$, respectively. The contour shown in Fig. S2a rotates clockwise, because the polar angle of W is equal to $22.5°$. The contour in Fig. S2b is drawn for the case when that angle is equal to $-22.5°$. This contour rotates counter-clockwise. These examples show that the right-hand rule is violated for the winding direction of chirp contours.

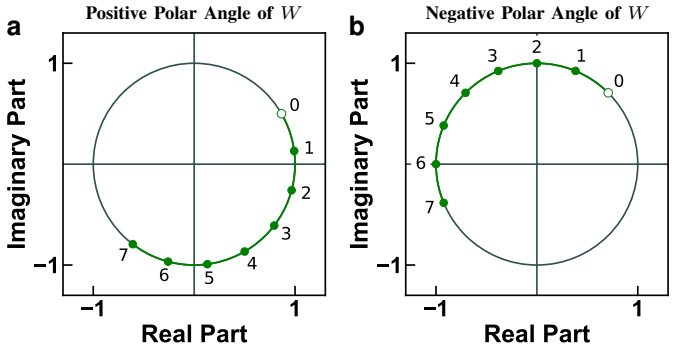

Fig. S2. Two 8-point chirp contours that lie on the unit circle: (**a**) for $A = \mathrm{e}^{\frac{i\pi}{6}}$ and $W = \mathrm{e}^{\frac{i\pi}{8}}$; and (**b**) for $A = \mathrm{e}^{\frac{i\pi}{4}}$ and $W = \mathrm{e}^{-\frac{i\pi}{8}}$. The contours are drawn according to Definition 6.

The chirp contour is shared by the CZT and the ICZT that have the same transform parameters. For the ICZT, however, there is no expression similar to Eq. (42) that uses the contour points to express the transform. Thus, for the ICZT it is more useful to view the contour as a visualization of the transform parameters A, W, and $M = N$.

S2. Mappings Between CZT, ICZT, FFT, and IFFT

This section shows how the CZT and ICZT algorithms can be used to compute the FFT and the IFFT for any N, i.e., not just a power of 2. Because orthogonal frequency components can be generated with either counter-clockwise or clockwise chirp contours, there are four possible algorithms. One of these mappings illustrates the ability to implement the IFFT using the CZT by reversing the contour direction and scaling the output vector. Some of the related work confused *reverse* with *inverse* and presented the resulting approach as the ICZT algorithm for all chirp contours that lie on the unit circle. The main paper shows why that approach is incorrect.

Algorithm S1 shows how to compute the FFT by calling the CZT algorithm with $M = N$, $W = \mathrm{e}^{-i2\pi/N}$, and $A = 1$. This works because these parameters define a Fourier contour with N points (see Section S1). Similarly, Algorithm S2 shows how to compute the IFFT by calling the ICZT. The parameter values in Algorithm S2 are the same as in Algorithm S1.

In contrast to the DFT, which uses a fixed contour that is traversed in only one direction, both the starting point and the contour direction can be varied with the CZT. This latter flexibility makes it possible to traverse the roots of unity in the clockwise direction and to compute the IFFT by calling the CZT. Algorithm S3 gives the pseudo-code for this approach. The CZT parameters in this case are $M = N$, $W = \mathrm{e}^{i2\pi/N}$, and $A = 1$. In addition, the algorithm scales all elements of the output vector by dividing them by N. This scaling is an integral part of the IFFT algorithm as well.

Similarly, the FFT can be implemented by calling the ICZT with $M = N$, $W = \mathrm{e}^{i2\pi/N}$, and $A = 1$. Again, the elements of the output vector need to be scaled. In this case, however, they are multiplied by N. Algorithm S4 gives the pseudo-code for this approach.

All four algorithms run in $O(n \log n)$, i.e., they have the same computational complexity as the standard FFT and IFFT algorithms. Their numerical accuracy, however, is somewhat lower than the numerical accuracy of the standard algorithms. The reason for this is that the CZT and the ICZT are more general algorithms that perform more operations, which increases the numerical error.

For the sake of completeness, Algorithm S5 gives the pseudo-code for the CZT algorithm. The pseudo-code for the ICZT is given in Algorithm 1. The dependencies for both algorithms are described in reference 10.

There is a peculiar extension of these mappings when the value of W is set to another *primitive* root of unity of order N instead of $\mathrm{e}^{-i2\pi/N}$ in Algorithms S1 and S2 or $\mathrm{e}^{i2\pi/N}$ in Algorithms S3 and S4. This leads to transforms that are still equivalent to the FFT or the IFFT after a cyclic permutation of the elements of the output vector.

The correspondence between these forward and inverse mappings breaks when the value of W is not a primitive root of unity of order N. For example, reversing the chirp contour computes the ICZT only if the frequency components are orthogonal, i.e., when the ICZT is equivalent to the IFFT after shuffling its output elements. This not so subtle issue was overlooked by some of the previous work. The main paper explains why this approach doesn't work.

Algorithm S1. FFT implemented using the CZT.

FFT-VIA-CZT($\mathbf{x}$)
$N \leftarrow$ LENGTH($\mathbf{x}$);
$\mathbf{X} \leftarrow$ CZT($\mathbf{x}$, N, $\mathrm{e}^{-\frac{i2\pi}{N}}$, 1);
return $\mathbf{X}$;

Algorithm S2. IFFT implemented using the ICZT.

IFFT-VIA-ICZT($\mathbf{X}$)
$N \leftarrow$ LENGTH($\mathbf{X}$);
$\mathbf{x} \leftarrow$ ICZT($\mathbf{X}$, N, $\mathrm{e}^{-\frac{i2\pi}{N}}$, 1);
return $\mathbf{x}$;

Algorithm S3. IFFT implemented using the CZT.

IFFT-VIA-CZT($\mathbf{X}$)
$N \leftarrow$ LENGTH($\mathbf{X}$);
$\mathbf{x} \leftarrow$ CZT($\mathbf{X}$, N, $\mathrm{e}^{\frac{i2\pi}{N}}$, 1);
for $j \leftarrow 0$ **to** $N - 1$ **do**
    $\mathrm{x}_j \leftarrow \dfrac{\mathrm{x}_j}{N}$;
end for
return $\mathbf{x}$;

Algorithm S4. FFT implemented using the ICZT.

FFT-VIA-ICZT($\mathbf{x}$)
$N \leftarrow$ LENGTH($\mathbf{x}$);
$\mathbf{X} \leftarrow$ ICZT($\mathbf{x}$, N, $\mathrm{e}^{\frac{i2\pi}{N}}$, 1);
for $j \leftarrow 0$ **to** $N - 1$ **do**
    $\mathrm{X}_j \leftarrow N \cdot \mathrm{X}_j$;
end for
return $\mathbf{X}$;

Algorithm S5. CZT Algorithm. Runs in $O(n \log n)$ time.

CZT($\mathbf{x}$, M, W, A)
$N \leftarrow$ LENGTH($\mathbf{x}$);
$\mathbf{X} \leftarrow$ EMPTYARRAY(N);
$\mathbf{r} \leftarrow$ EMPTYARRAY(N);
$\mathbf{c} \leftarrow$ EMPTYARRAY(M);
for $k \leftarrow 0$ **to** $N - 1$ **do**
    $\mathrm{X}[k] \leftarrow W^{\frac{k^2}{2}} \cdot A^{-k} \cdot \mathrm{x}[k]$;
    $\mathrm{r}[k] \leftarrow W^{-\frac{k^2}{2}}$;
end for
for $k \leftarrow 0$ **to** $M - 1$ **do**
    $\mathrm{c}[k] \leftarrow W^{-\frac{k^2}{2}}$;
end for
// After the next line, LENGTH($\mathbf{X}$) $= M$.
$\mathbf{X} \leftarrow$ TOEPLITZMULTIPLYE($\mathbf{r}$, $\mathbf{c}$, $\mathbf{X}$);
for $k \leftarrow 0$ **to** $M - 1$ **do**
    $\mathrm{X}[k] \leftarrow W^{\frac{k^2}{2}} \cdot \mathrm{X}[k]$;
end for
return $\mathbf{X}$;

S3. Implementing ICTA and IFRFT using ICZT

The main paper mentioned two algorithms that generalize the *Fast Fourier Transform* (FFT) on the unit circle. They are the *Chirp Transform Algorithm* (CTA) and the *Fractional Fourier Transform* (FRFT) algorithm, which are described in references 15 and 16. The *Chirp Z-Transform* (CZT) is a generalization of the FFT off the unit circle. As described in this paper, however, the CZT can also be evaluated for circular chirp contours that lie on the unit circle. Thus, the CZT can also be viewed as a generalization of the FFT on the unit circle. Interestingly, both the CTA and the FRFT can be implemented with the CZT as shown below.

The corresponding inverse algorithms have not been discovered yet and have not been described in the literature until now. Both of these algorithms can be stated as special cases of the Inverse Chirp Z-Transform (ICZT) algorithm as described in this section. We took the liberty of naming these algorithms by adding an 'I' as the first letter in both acronyms.

Algorithm S6. Chirp Transform Algorithm (CTA).

1: CTA($\mathbf{x},\ M,\ \omega_0,\ \Delta\omega$)
2: $(W,\ A) \leftarrow (\mathrm{e}^{-i\,\Delta\omega},\ \mathrm{e}^{i\omega_0})$;
3: $\mathbf{X} \leftarrow$ CZT($\mathbf{x},\ M,\ W,\ A$);
4: **return** $\mathbf{X}$;

Algorithm S6 shows how to implement the CTA using the CZT. This is done by mapping the parameters of the CTA to the parameters of the CZT. The first two parameters are identical for both algorithms. They specify the input vector, $\mathbf{x}$, and the size of the output vector, M. The CTA defines the starting point of the contour using the starting angle ω_0. The angular distance between adjacent contour points is given by the parameter $\Delta\omega$. Line 2 of Algorithm S6 computes the CZT parameters W and A from the CTA parameters ω_0 and $\Delta\omega$. Line 3 calls the CZT algorithm to compute the output vector, which is returned on line 4. The pseudo-code for the CZT algorithm is given in Algorithm S5. Reference 10 gives the pseudo-code for all of its dependencies.

Historically, the chirp contour points were defined[7] using the z-transform instead of the power series (i.e., negative powers instead of positive powers, as described in Section S1). The CZT was also defined[7] using negative powers of the parameter A. More formally, the chirp contour points are set to the M complex numbers $\mathrm{z}_0, \mathrm{z}_1, \ldots, \mathrm{z}_{M-1}$ where $\mathrm{z}_k = A\,W^{-k}$. The value of the k-th element of the CZT output vector $\mathbf{X}$ is equal to the value of the z-transform at z_k of the CZT input vector $\mathbf{x}$, i.e.,

$$\mathrm{X}_k = \sum_{j=0}^{N-1} \mathrm{x}_j\, \mathrm{z}_k^{-j}, \quad \text{where } k \in \{0, 1, \ldots, M-1\}. \tag{46}$$

Following these traditions requires setting W to $\mathrm{e}^{-i\,\Delta\omega}$ and A to $\mathrm{e}^{i\omega_0}$ when mapping the CTA to the CZT. In other words, a positive value of ω_0 corresponds to a counter-clockwise offset for the starting point of the chirp contour relative to the point $(1, 0)$ on the unit circle. Similarly, a positive value of $\Delta\omega$ corresponds to a counter-clockwise winding direction of the chirp contour[15].

Algorithm S7. Fractional Fourier Transform (FRFT).

1: FRFT($\mathbf{x},\ m,\ \alpha$)
2: $(M,\ W,\ A) \leftarrow (m,\ \mathrm{e}^{-i2\pi\alpha},\ 1)$;
3: $\mathbf{G} \leftarrow$ CZT($\mathbf{x},\ M,\ W,\ A$);
4: **return** $\mathbf{G}$;

Algorithm S7 gives the pseudo-code for implementing the FRFT using the CZT. Once again, this is done by mapping the parameters of the FRFT to the parameters of the CZT. In this case, $\mathbf{x}$ is the input vector, which is the same for both algorithms. The second parameter is m, which is equivalent to M in the CZT formulation. The last parameter, α, specifies the angular distance between two adjacent contour points. It maps to the parameter W through the formula $W = \mathrm{e}^{-i2\pi\alpha}$. By definition, all FRFT contours start at 1. Thus, the CZT parameter A, which specifies the starting point of the contour, is always set to 1. The CTA is more flexible in this respect, because it allows the starting point of the contour to be any point on the unit circle, i.e., by varying the parameter ω_0.

Both the CTA and the FRFT can be used with contours that perform more than one revolution on the unit circle. Both algorithms run in $O(n \log n)$ time.

Algorithm S8. Inverse Chirp Transform Algorithm (ICTA).

1: ICTA($\mathbf{X},\ N,\ \omega_0,\ \Delta\omega$)
2: $(W,\ A) \leftarrow (\mathrm{e}^{-i\,\Delta\omega},\ \mathrm{e}^{i\omega_0})$;
3: $\mathbf{x} \leftarrow$ ICZT($\mathbf{X},\ N,\ W,\ A$);
4: **return** $\mathbf{x}$;

Algorithm S9. Inverse Fractional Fourier Transform (IFRFT).

1: IFRFT($\mathbf{G},\ m,\ \alpha$)
2: $(\mathbf{X},\ N,\ W,\ A) \leftarrow (\mathbf{G},\ m,\ \mathrm{e}^{-i2\pi\alpha},\ 1)$;
3: $\mathbf{x} \leftarrow$ ICZT($\mathbf{X},\ N,\ W,\ A$);
4: **return** $\mathbf{x}$;

Algorithms S8 and S9 show how to implement the ICTA and the IFRFT by mapping their parameters to the ICZT parameters and then calling Algorithm 1. Algorithm S8 computes the values of W and A for the ICZT from the parameters ω_0 and $\Delta\omega$ and then calls Algorithm 1 to compute the transform. Similarly, Algorithm S9 computes the ICZT parameters from $\mathbf{G}$, m, and α and then also calls Algorithm 1. Both algorithms run in $O(n \log n)$ time and use $O(n)$ memory.

By definition, the ICTA and the IFRFT algorithms can work only with contours that lie on the unit circle. The ICZT algorithm, however, can also work with logarithmic spiral contours that lie off the unit circle[10]. Thus, the ICZT is more general than both the ICTA and the IFRFT.

S4. The Rank of the Matrix $\boldsymbol{W}$ Depends on the Polar Angle of the Transform Parameter W

This section provides additional proofs that complement the results from Theorem 2. In this case, the analysis is performed using the rank of the matrix $\boldsymbol{W}$ in Eq. (4) instead of the singularities of Eq. (11) that specifies the generating vector $\mathbf{u}$.

Theorem 3. *Let p/q be a rational number represented as an irreducible fraction, i.e., p is an integer and q is a positive integer such that $\gcd(p,q)=1$. Furthermore, let n be a positive integer. Then, the number of elements in the set $S=\left\{\mathrm{e}^{i2\pi\frac{p\cdot 0}{q}},\mathrm{e}^{i2\pi\frac{p\cdot 1}{q}},\ldots,\mathrm{e}^{i2\pi\frac{p\cdot(n-1)}{q}}\right\}$ is equal to the smaller of q and n. More formally,*

$$\left|\left\{\mathrm{e}^{i2\pi\frac{pk}{q}},\ k\in\{0,1,\ldots,n{-}1\}\right\}\right|=\min(q,n). \tag{47}$$

Proof. Let k_1 and k_2 be two different integers that lie between 0 and $\min(q,n){-}1$. That is,

$$k_1,k_2\in\{0,1,2,\ldots,\min(q,n){-}1\},\quad k_1\neq k_2. \tag{48}$$

Then, the difference between their corresponding elements in the set S can be expressed as follows:

$$\mathrm{e}^{i2\pi\frac{p}{q}k_1}-\mathrm{e}^{i2\pi\frac{p}{q}k_2}=\mathrm{e}^{i2\pi\frac{p}{q}k_1}\big(1-\mathrm{e}^{i2\pi\frac{p}{q}(k_2-k_1)}\big). \tag{49}$$

Without loss of generality, suppose that $k_1<k_2$. Then, the following inequality holds:

$$1\leq k_2-k_1\leq\min(q,n)-1<q. \tag{50}$$

Because $\gcd(p,q)=1$, it follows that p and q share no prime factors. Therefore,

$$\gcd(p(k_2-k_1),q)=\gcd(k_2-k_1,q)\leq k_2-k_1<q. \tag{51}$$

That is, each prime factor shared by the product $p(k_2-k_1)$ and q must be shared by k_2-k_1 and q. Moreover, the value of $\gcd(k_2-k_1,q)$ can't exceed k_2-k_1, which is strictly less than q. This implies that the fraction $\frac{p(k_2-k_1)}{q}$ is not reducible to an integer. Hence,

$$\mathrm{e}^{i2\pi\frac{p}{q}(k_2-k_1)}\neq 1, \tag{52}$$

which implies that:

$$\mathrm{e}^{i2\pi\frac{p}{q}k_1}\neq\mathrm{e}^{i2\pi\frac{p}{q}k_2}. \tag{53}$$

Therefore, the mapping from $k\in\{0,1,2,\ldots,\min(q,n)-1\}$ to $\mathrm{e}^{i2\pi\frac{p}{q}k}$ is one-to-one. Thus, there are at least $\min(q,n)$ distinct elements in the set S, which proves that $|S|\geq\min(q,n)$.

To prove the equality in Eq. (47), it only remains to show that $|S|\leq\min(q,n)$. There are two possible cases: $n\leq q$ and $n>q$. If $n\leq q$, then $|S|\leq n=\min(q,n)$ because, by definition, the number of elements in S cannot exceed n.

If $n>q$, then for each $k\geq q$ the value of $\mathrm{e}^{i2\pi\frac{p}{q}k}$ is equal to $\mathrm{e}^{i2\pi\frac{p}{q}(k-q)}$. In other words, the elements begin to repeat starting with $k=q$. More formally,

$$\mathrm{e}^{i2\pi\frac{p(k-q)}{q}}=\mathrm{e}^{i2\pi\frac{pk}{q}}\underbrace{\mathrm{e}^{-i2\pi p}}_{1}=\mathrm{e}^{i2\pi\frac{pk}{q}}, \tag{54}$$

for each $k\geq q$. Thus, $|S|\leq q=\min(q,n)$. □

Theorem 4. *Let p/q be a rational number that is represented using an irreducible fraction, i.e., p is an integer and q is a positive integer such that $\gcd(p,q)=1$. Let M and N be two positive integers. Also, let $\boldsymbol{W}$ be the M-by-N Vandermonde matrix used by the CZT where $W=\mathrm{e}^{i2\pi p/q}$. That is,*

$$\boldsymbol{W}=\begin{bmatrix}W^{0\cdot 0} & W^{1\cdot 0} & \ldots & W^{(N-1)\cdot 0}\\ W^{0\cdot 1} & W^{1\cdot 1} & \ldots & W^{(N-1)\cdot 1}\\ \vdots & \vdots & \ddots & \vdots\\ W^{0\cdot(M-1)} & W^{1\cdot(M-1)} & \ldots & W^{(N-1)\cdot(M-1)}\end{bmatrix}. \tag{55}$$

Then,

$$\operatorname{rank}(\mathbf{W})=\min(q,n),\quad \textit{where } n=\min(M,N). \tag{56}$$

Proof. Without loss of generality, suppose that $M\leq N$, which implies that $n=M$ (otherwise, if $M>N$, then the matrix $\boldsymbol{W}$ can be replaced with its transpose, which doesn't affect the rank). The matrix $\boldsymbol{W}$ is a Vandermonde matrix that is generated by the vector $\mathbf{s}=(W^0,W^1,W^2,\ldots,W^{n-1})$. The number of distinct elements in this vector is equal to $\min(q,n)$, which follows from Theorem 3. Thus, the number of distinct rows in the matrix $\boldsymbol{W}$ is also equal to $\min(q,n)$. This implies that $\operatorname{rank}(\boldsymbol{W})\leq\min(q,n)$.

To prove the equality in Eq. (56) we will show that the first r rows of $\boldsymbol{W}$ are linearly independent, where $r=\min(q,n)$. Let $\boldsymbol{V}$ be a square sub-matrix of $\boldsymbol{W}$ that lies in the intersection of its first r rows and r columns. The matrix $\boldsymbol{V}$ is also a Vandermonde matrix that is generated by the vector $(W^0,W^1,W^2,\ldots,W^{r-1})$. Theorem 3 proved that the number of distinct elements in this vector is equal to r.

The determinant of $\boldsymbol{V}$ is given by the following formula (see reference 39, p. 191 or reference 43, p. 78):

$$\det(\boldsymbol{V})=\prod_{0\leq i<j\leq n-1}(W^j-W^i)\neq 0. \tag{57}$$

This implies that the matrix $\boldsymbol{V}$ is non-singular, i.e, that it has full rank. More formally, $\operatorname{rank}(\boldsymbol{V})=r$. Because $\boldsymbol{V}$ is a sub-matrix of $\boldsymbol{W}$, it follows that

$$r=\operatorname{rank}(\boldsymbol{V})\leq\operatorname{rank}(\boldsymbol{W}). \tag{58}$$

Thus, Eq. (56) holds because $r\leq\operatorname{rank}(\boldsymbol{W})\leq r$. □

The following theorem reinterprets Theorem 4 in terms of Farey sequences. It shows that the matrix $\boldsymbol{W}$ is singular when the polar angle of the transform parameter W is a Farey angle of order $n-1$, where $n=M=N$ is the size of the transform.

Theorem 5. *The matrix $\mathbf{W}$ is singular when $p/q\in F_{n-1}$, i.e., p/q is a member of the Farey sequence of order $n-1$. Conversely, if $p/q\notin F_{n-1}$, then $\mathbf{W}$ is non-singular. Another way to state these conditions is: $\mathbf{W}$ is singular when $q<n$ and nonsingular when $q\geq n$, where q is a positive integer, $p\in\{0,1,\ldots,q\}$, $p/q\in[0,1]$, $\gcd(p,q)=1$, and $W=\mathrm{e}^{i2\pi p/q}$.*

Proof. The definition of a Farey sequence implies that all rational numbers p/q between 0 and 1 for which $q<n$ are elements of F_{n-1}. Thus, $\mathbf{W}$ is singular whenever $p/q\in F_{n-1}$. Conversely, if $p/q\notin F_{n-1}$, then $q\geq n$ and $\min(q,n)=n$, which implies that the matrix $\boldsymbol{W}$ is non-singular. □

Theorem 6. *Let F_n be the Farey sequence of order n. Then, the sequence $S = \left\{ \mathrm{e}^{i2\pi p/q} : {}^{p}/_{q} \in F_n \right\}$ lists each complex root of unity of order $1, 2, \ldots, n$ exactly once. Let Ω be the set of these roots of unity. More formally,*

$$\Omega = \{\omega_b^a \,:\, b \in \{1, 2, \ldots, n\},\ a \in \{0, 1, 2, \ldots, b{-}1\}\}, \tag{59}$$

where $\omega_b^a = \mathrm{e}^{i2\pi a/b}$. Then, $\Omega = S$.

Proof. Let ${}^{p}/_{q}$ be an element of F_n. Then, $q \in \{1, 2, \ldots, n\}$, and $p \in \{0, 1, \ldots, q\}$, which implies $(\mathrm{e}^{i2\pi p/q})^q = \mathrm{e}^{i2\pi p} = 1$. Thus, $\mathrm{e}^{i2\pi p/q}$ is a q-th root of unity. Therefore, $\mathrm{e}^{i2\pi p/q} \in \Omega$. From this it follows that $S \subseteq \Omega$.

Conversely, let $\omega_b^a \in \Omega$. Then, $\omega_b^a = \mathrm{e}^{i2\pi a/b}$, where $a \in \{0, 1, 2, \ldots, b{-}1\}$ and $b \in \{1, 2, \ldots, n\}$. Let p and q be two integers defined as: $p = a/\gcd(a, b)$ and $q = b/\gcd(a, b)$. Because $\gcd(a, b) \geq 1$, it follows that $1 \leq q \leq b \leq n$. Because $a < b$ and p and q are obtained by dividing a and b with the same positive number, it follows that $0 < p < q \leq n$. Thus, ${}^{p}/_{q} \in F_n$. Therefore, $\mathrm{e}^{i2\pi a/b} = \omega_b^a = \mathrm{e}^{i2\pi p/q} \in S$, from which it follows that $\Omega \subseteq S$.

Combining the two results leads to $\Omega = S$. □

The next theorem proves that using a chirp contour with sampling points that coincide with the sampling points used by the FFT, i.e., a chirp contour with $W = \mathrm{e}^{i2\pi k/n}$ where k is coprime with n, never leads to singularities in the CZT matrix. This is done by showing that, in this special case, the fraction ${}^{k}/_{n}$ appears for the first time in the Farey sequence F_n. The singularities of the CZT matrix, however, are determined by the elements of the preceding Farey sequence F_{n-1}. Thus, the singularities are avoided when the CZT reduces to the FFT. The same argument applies to the ICZT and the IFFT.

Theorem 7. *Let n be a positive integer and let $\omega_n^k = \mathrm{e}^{i2\pi \frac{k}{n}}$ be a primitive root of unity of order n, i.e., ω_n^k is not a root of unity of any order smaller than n. Then, the following two conditions hold:*

1) *the fraction ${}^{k}/_{n}$ is an element of the Farey sequence of order n, i.e,. ${}^{k}/_{n} \in F_n$;*
2) *the fraction ${}^{k}/_{n}$ is not an element of the Farey sequence of order $n{-}1$, i.e., ${}^{k}/_{n} \notin F_{n-1}$.*

Proof. Without loss of generality, we can assume that k is an integer between 0 and $n{-}1$ (otherwise, if $k \geq n$ or $k < 0$, then we can set k to $k - \left\lfloor \frac{k}{n} \right\rfloor n$, which is between 0 and $n{-}1$, without changing the value of $\mathrm{e}^{i2\pi k/n}$). Therefore, by definition, the fraction ${}^{k}/_{n}$ is an element of the Farey sequence of order n, i.e., ${}^{k}/_{n} \in F_n$.

The second condition is proven by contradiction. Suppose that ${}^{k}/_{n} \in F_{n-1}$. This implies that the fraction ${}^{k}/_{n}$ is reducible, i.e., ${}^{k}/_{n} = {}^{p}/_{q}$, where q is a positive integer smaller than n and p is an integer between 0 and $q{-}1$. In turn, this reducibility implies that ω_n^k is not a *primitive* root of unity of order n because it is a root of unity of order q where $q < n$. That is,

$$(\omega_n^k)^q = \mathrm{e}^{i2\pi \frac{k}{n} q} = \mathrm{e}^{i2\pi \frac{p}{q} q} = \mathrm{e}^{i2\pi p} = 1. \tag{60}$$

This contradiction proves that ${}^{k}/_{n} \notin F_{n-1}$, as required. □

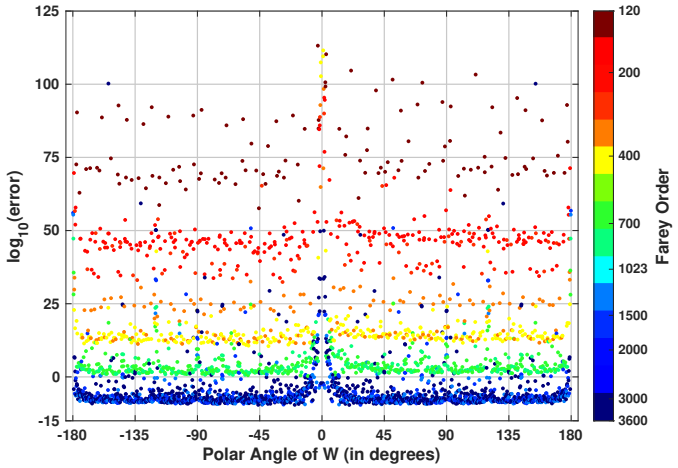

Fig. S3. Explanation of the layering in Fig. 6a in terms of the Farey order of each angle used in this discretization.

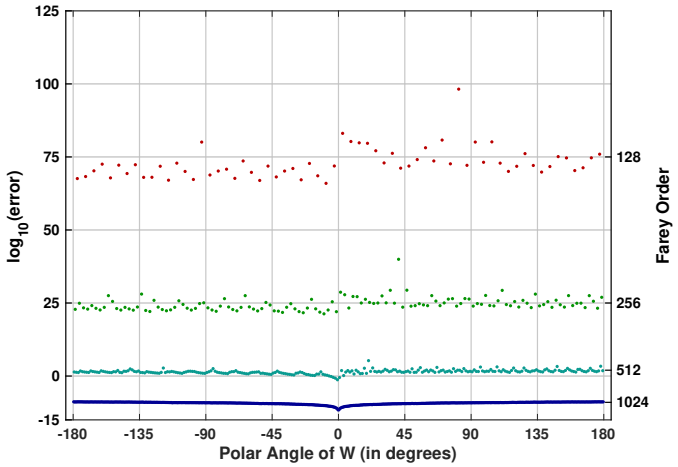

Fig. S4. Another example with color-coded layers, this time from Fig. 7a, where both the transform size and the number of regularly-discretized polar angles were set to 1024.

Figure S3 illustrates Theorem 4 and Theorem 5. It shows the same plot as in Fig. 6a but also colors each point based on the Farey order of its corresponding polar angle of W. The colors nicely explain the stratification of the error function for this discretization, which uses a step of $0.1°$. The upper layers, which have very large errors, correspond to lower Farey orders. The lower layers correspond to larger Farey orders. In general, the transform is numerically accurate for Farey orders that exceed or are equal to the transform size. In this plot they are drawn with blue colors.

Figure S4 gives another illustration of these effects using a color version of the plot shown in Fig. 7a. In this case the Farey orders can only be integer powers of two, because both the number of regularly-discretized angles and the transform size are equal to 1024. The four layers in this plot correspond to Farey orders 128, 256, 512, and 1024. The numerical error for Farey orders less than 128 is too large for computations with double precision, leading to either `NaN`[25] or infinite numerical error values for those polar angles of W.

S5. Additional Results for 32-Point Chirp Contours

This section studies the behavior of the numerical error for 32-point chirp contours on the unit circle. The goal is to determine if the harmonically spaced spiking pattern shown in Fig. 5a (i.e., the "harmonic hedgehog") can be observed for these contours as well.

Figure S5 plots the absolute numerical error for 32-point contours as a function of the polar angle of the transform parameter W. The angles were discretized using a 0.1° step. The overall shape of the numerical error function is similar to the error shown in Fig. 4, but the pattern becomes more complicated. For example, the features at the bottom of the figure are less distinct at this resolution and appear to be compressed by a factor of 2.

Figure S6 shows a close-up view of the numerical error for angles between 10° and 23°. The discretization interval in this case is 0.005°. The figure reveals another spiking pattern. In this case, however, there are 16 spikes instead of 8. Also, the Farey angles in this case are related to F_{31} instead of F_{15}. Furthermore, the harmonic hedgehog is now squeezed in the interval $[11^\circ, 23^\circ]$ instead of the interval $[22^\circ, 46^\circ]$ as in Fig. 5a. In addition to the discretized angles, the plot also includes points that correspond to Farey angles in the set $\{\frac{360^\circ}{31}, \frac{360^\circ}{30}, \frac{360^\circ}{29}, \ldots, \frac{360^\circ}{16}\}$. The vertical coordinate exceeds zero only for these sixteen Farey angles.

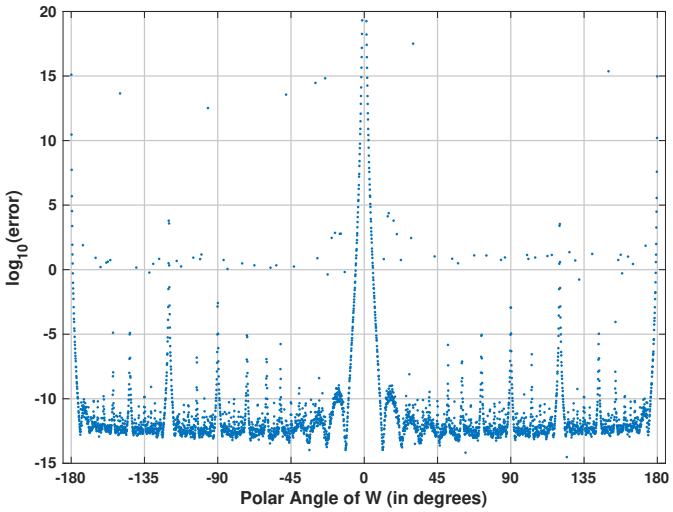

Fig. S5. Absolute numerical error of the CZT–ICZT procedure for chirp contours with 32 points on the unit circle. The discretization step for the polar angles of W was set to 0.1°, which resulted in 3600 angles.

The red points in Fig. S6 correspond to chirp contours with polar angles equal to 22.2°, 22.5°, and 22.8° (see Fig. S7). The middle red point coincides with a spike in the numerical error because its corresponding contour has only 16 distinct sampling points. That is, the contour makes exactly 2 revolutions and the first 16 points coincide with the second 16 points. The CZT transformation matrix for this contour is singular, i.e., it is not invertible. Section S4 proves that the matrix $\boldsymbol{W}$ (see Eq. (4)) is not full rank when the polar angle of W is a Farey angle of order q and q is strictly less than the transform

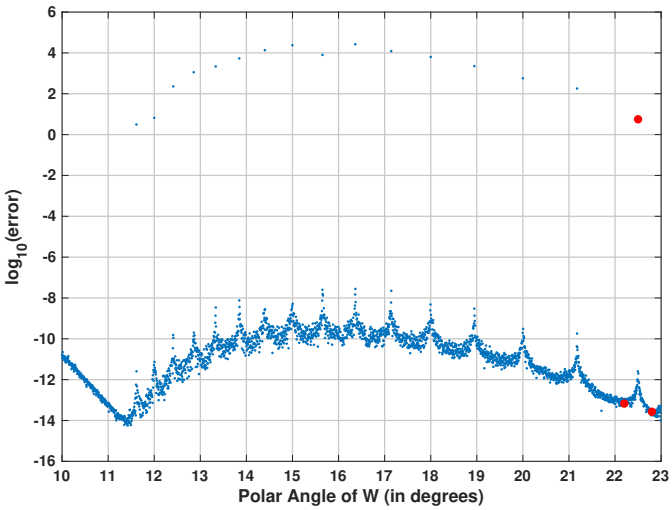

Fig. S6. The harmonic hedgehog for chirp contours with 32 points has 16 spikes. The absolute numerical error spikes when the polar angle of W is close to an element of the set $\{\frac{360^\circ}{31}, \frac{360^\circ}{30}, \frac{360^\circ}{29}, \ldots, \frac{360^\circ}{16}\}$. The three red points indicate the error for each of the three contours shown in Fig. S7.

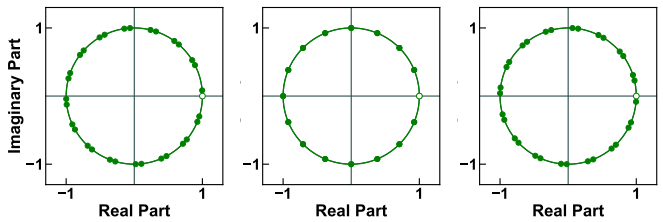

Fig. S7. Three chirp contours, each with 32 points, that wrap twice around the unit circle. From left to right, the polar angle of W is: 22.2°, 22.5°, and 22.8°. The middle contour has only 16 distinct points because 22.5° is equal to $360^\circ/16$. In other words, the 16 points from the second revolution coincide with the 16 points from the first revolution.

size n. In this case, $q = 16$ and $n = 32$, which explains why the error spikes near this angle in Fig. S6.

The error for the other two contours from Fig. S7 is relatively small. Each of them has 16 pairs of points that are very close to each other but do not coincide. The left contour corresponds to a Farey angle of order 600. The right contour corresponds to a Farey angle of order 300. In contrast, the middle contour corresponds to a Farey angle of order 16, which is below the transform size that is equal to 32.

Once again, the results indicate that the numerical error is very small for most contours that were tried. The only exceptions are contours for which the polar angle of W is close to a Farey angle or 0°, which is also a Farey angle. In theory, Theorem 2 from the main paper suggests that the error should be infinite when the polar angle of W is a Farey angle of order less than n. In practice, however, the error is often large but finite for these angles. For example, even though sixteen Farey angles were explicitly added in Fig. S6, the absolute numerical error for these points is still between 10^0 and 10^5. This is due to the inability of the IEEE-754 floating-point representation[25], which is used by modern computers, to represent these Farey angles and their complex exponentials exactly.

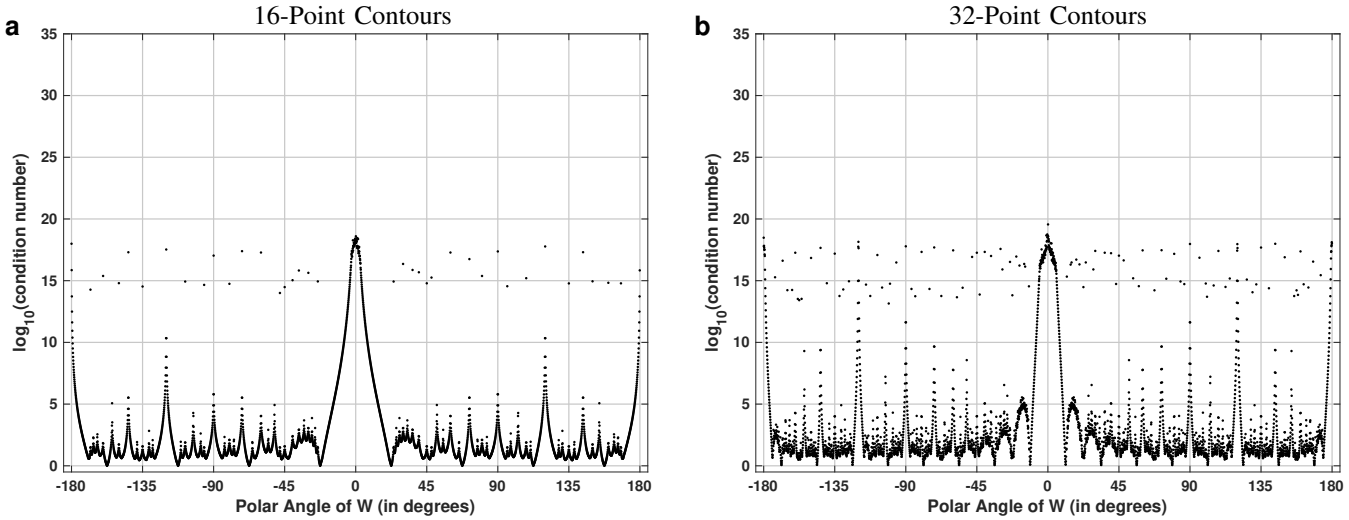

Fig. S8. Condition numbers for the transform matrix $\boldsymbol{W}$, shown as a function of the polar angle of the transform parameter W for chirp contours on the unit circle with 16 points in (**a**) and 32 points in (**b**). For both plots, the polar angles were discretized at 0.1° intervals, for a total of 3600 regularly-spaced angles.

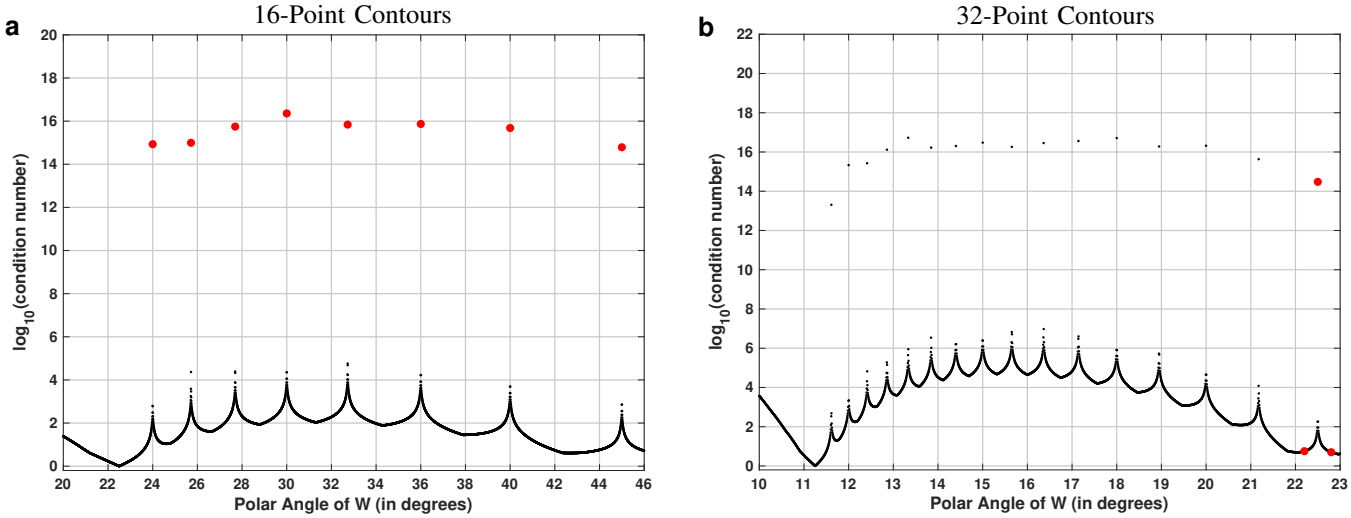

Fig. S9. (**a**) Close-up of the harmonic hedgehog between 20° and 46° in Fig. S8a. The eight red points indicate the condition numbers at the Farey angles in the set $\{\frac{360^\circ}{15}, \frac{360^\circ}{14}, \frac{360^\circ}{13}, \ldots, \frac{360^\circ}{8}\}$. (**b**) Close-up of the harmonic hedgehog between 10° and 23° in Fig. S8b. The three red points indicate the condition numbers for each of the three contours from Fig. S7. The condition number spikes near Farey angles in the set $\{\frac{360^\circ}{31}, \frac{360^\circ}{30}, \frac{360^\circ}{29}, \ldots, \frac{360^\circ}{16}\}$. The shapes of these plots are similar to the shapes of the numerical error plots in Fig. 5a and Fig. S6, respectively.

S6. Condition Numbers for 16-Point and 32-Point Chirp Contours

Figure S8 plots the condition number of the matrix $\boldsymbol{W}$ from Eq. (4) as a function of the polar angle of the parameter W for 16-point and 32-point contours. For both plots, the discretization used 3600 regularly-spaced polar angles between 0° and 360°, i.e., in increments of 0.1°. The condition numbers were computed with 64-bit floating-point numbers as follows: $\mathrm{cond}(\boldsymbol{W}) = \sigma_{\max}/\sigma_{\min}$, where $\sigma_{\max}$ is the maximum and $\sigma_{\min}$ is the minimum singular value of the matrix $\boldsymbol{W}$.

For 16-point contours, the shape of the condition number plot in Fig. S8a is similar to the shapes of the predicted and empirical numerical error plots shown in Fig. 4 in the main paper. For 32-point contours, the shape of the plot in Fig. S8b is similar to the shape of the numerical error plot in Fig. S5. The locations of the peaks in Figure S8 are in agreement with the theoretical results from Section S4.

Figure S9 shows close-ups of two harmonic hedgehogs. Fig. S9a zooms in on the interval $[20^\circ, 46^\circ]$ from Fig. S8a. Fig. S9b focuses on the interval $[10^\circ, 23^\circ]$ from Fig. S8b. In both plots, the discretization interval is 0.005°.

The red points in Fig. S9a indicate the condition numbers for the Farey angles that correspond to the elements of the Farey sequence F_{15} between $1/15$ and $1/8$. These angles were explicitly added because they were missed by this discretization. Similarly, the points that correspond to the elements of F_{31} between $1/31$ and $1/16$ were included in Fig. S9b. The three red points in that figure indicate the condition numbers for the three 32-point contours from Fig. S7.

The shapes of the two harmonic hedgehogs in Fig. S9 are similar to the numerical error plots in Fig. 5a and Fig. S6. This suggests that the pattern of harmonically-spaced spikes can be observed not only with the numerical error but also with the condition number for this problem.

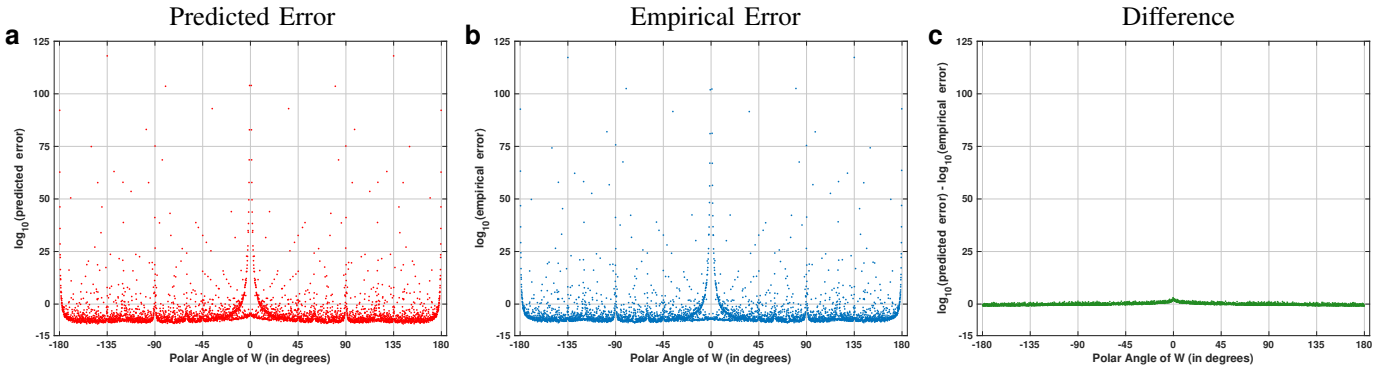

Fig. S10. Visualization of the predictive accuracy of Eq. (21) for 2048-point chirp contours. The red points in (**a**) show the predicted numerical error for the CZT followed by the ICZT, computed for 4099 regularly-discretized polar angles of W. The blue points in (**b**) show the empirically-observed numerical error, which is averaged over 10 random input vectors. The green points in (**c**) show the difference between the predicted error and the empirically observed error, i.e., (**a**) − (**b**).

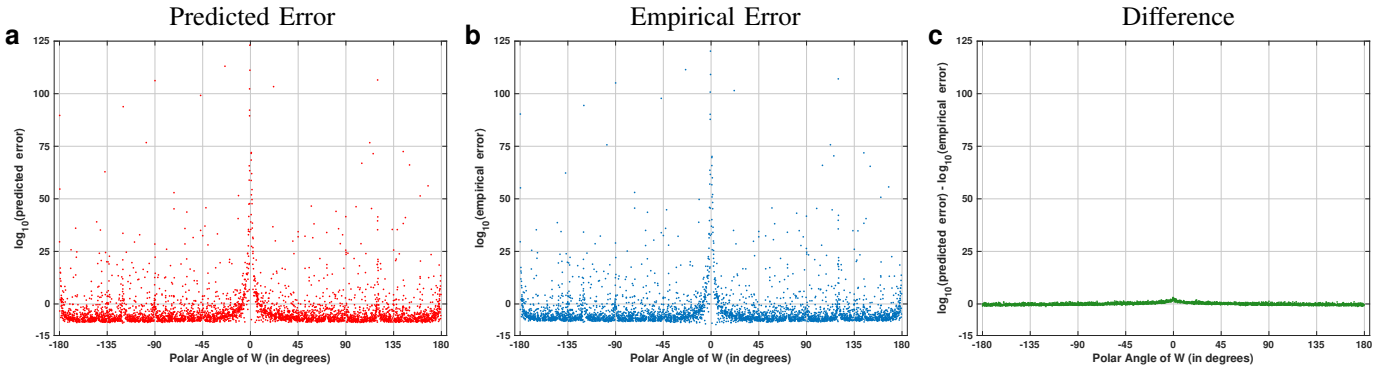

Fig. S11. Visualization of the accuracy of the error prediction formula for the CZT followed by the ICZT. These are similar to the plots in Fig. S10, but in this case the results are shown for 4099 randomly-sampled polar angles of W.

S7. Additional Error Analysis Results

Figure S10 visualizes the accuracy of the numerical error predictions obtained with Eq. (21). The results are plotted for 2048-point chirp contours and for 4099 regularly-discretized polar angles of W. The red points in Fig. S10a show the predicted numerical error. The blue points in Fig. S10b show the average empirically-observed error. The green points in Fig. S10c show the difference between the predicted error and the observed error, i.e, red minus blue. Figure S11 shows a similar analysis for randomly-sampled polar angles.

Figures S10c and S11c plot the difference between the logarithms of the predicted numerical error and the empirical numerical error. In both cases, the predicted error is close to the empirical error, which is reflected in the green horizontal line at zero. The slight bump around 0° in both figures suggests that the error prediction formula slightly overestimates the numerical error for polar angles of W that are close to 0°. Because the R^2 coefficient tends to 1 as additional points are added to the chirp contour, the impact of this overestimation effect relative to the total variance of the numerical error diminishes as N increases.

For discretizations that hit many Farey angles, both the predicted and the observed errors could be very large (e.g., see Fig. 6a). For some of these singularities, the formulas described in the main paper underestimate the error. Section S8 analyzes these special cases in more detail and shows how to patch the numerical error prediction formulas by modifying the offset term B when the value of W is close to an ICZT singularity. This change makes the predictions more accurate for these extreme cases, but those values of W should not be used in practice because even the original formulas predict a very large error. This patch was not used for Figs. S10 and S11.

All algorithms described in this paper were implemented in C++. For this implementation, the constants C_1 and C_2 in Eq. (19) have the following values: $C_1 = 1$ and $C_2 = -1$. In our previous paper[10], the algorithms were implemented in Python using the *mpmath* library (see http://mpmath.org/), which slightly boosts the precision of complex exponentiation. There the precision was further boosted when computing all generating vectors for the matrices used by the CZT and the ICZT. This further decreased the numerical error so that the value of C_1 for that implementation reduced to -1 and the value of C_2 became 0. These implementation-specific details affect only the offset term B, i.e., they only shift the error function up or down by a fixed amount without affecting its overall shape.

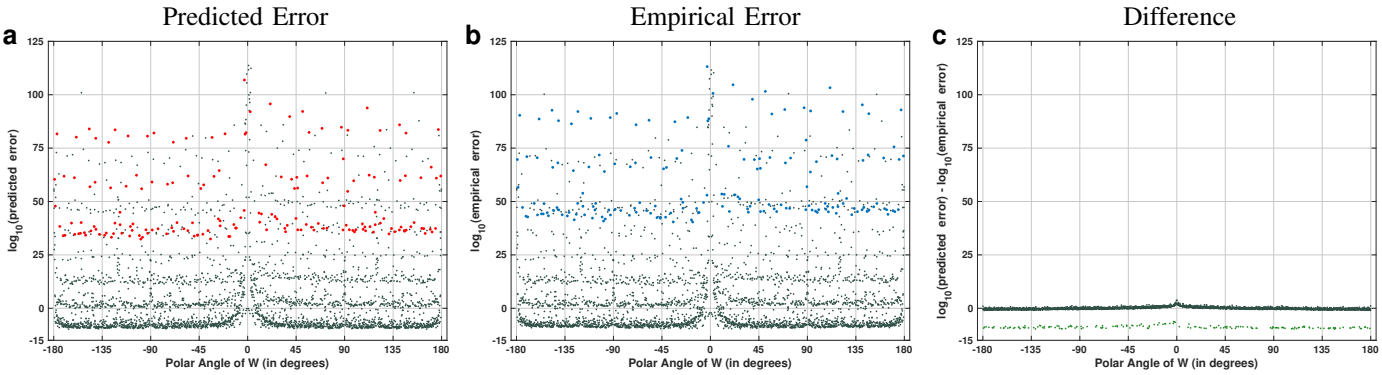

Fig. S12. Example in which the numerical error is sometimes underestimated. The predicted numerical error is shown in (**a**). The empirically-observed error is shown in (**b**) – this is the same plot as in Fig. 6a. The difference between (**a**) and (**b**) is shown in (**c**). The green points in (**c**) indicate all cases for which Eq. (19) and Eq. (21) underestimate the numerical error. The red points in (**a**) and the blue points in (**b**) show the predicted and the empirically-observed errors for these green points.

S8. Error Analysis Near ICZT Singularities

This section describes a slight modification for the error prediction formulas that makes them more accurate near ICZT singularities. The formulas described in the main paper can underestimate the predicted error for some of these cases. More specifically, the offset term B given by Eq. (19), which is used in Eq. (21) and Eq. (23), is modified here to be more accurate near ICZT singularities. This modification is not necessary in practice because even the underestimated error value is already very high.

Figure S12 shows an example for a regular discretization with a step of 0.1° that hits many singularities. For some of them the predicted numerical error is underestimated, as indicated by the green points below zero in Fig. S12c. The empirically-observed errors for these green points are highlighted in blue in Fig. S12b. The corresponding predicted errors are highlighted in red in Fig. S12a. The error is underestimated because the red points appear lower than the blue points.

The modified offset term B is computed using three helper terms: S_1, S_2, and S_3. The terms S_1 and S_2 are equal to the logarithms of the norms of the vectors $\mathbf{x}'$ and $\mathbf{x}''$ that are computed by Algorithm 1 on lines 25–28, i.e.,

$$S_1 = \log \|\mathbf{x}'\| = \log \sqrt{\sum_{k=0}^{N-1} |\mathrm{x}'_k|^2} = \frac{1}{2} \log \sum_{k=0}^{N-1} |\mathrm{x}'_k|^2, \quad (61)$$

$$S_2 = \log \|\mathbf{x}''\| = \log \sqrt{\sum_{k=0}^{N-1} |\mathrm{x}''_k|^2} = \frac{1}{2} \log \sum_{k=0}^{N-1} |\mathrm{x}''_k|^2. \quad (62)$$

The term S_3 is equal to the logarithm of the Euclidean distance between $\mathbf{x}'$ and $\mathbf{x}''$. That is,

$$S_3 = \log \|\mathbf{x}' - \mathbf{x}''\| = \frac{1}{2} \log \sum_{k=0}^{N-1} |\mathrm{x}'_k - \mathrm{x}''_k|^2. \quad (63)$$

Near ICZT singularities the difference between S_3 and the larger value between S_1 and S_2 captures the numerical error offset better than the constant term $-p \log 2$ used in Eq. (19).

Let p/q be the rational approximation of the polar angle of W expressed as a fraction of a turn, i.e., $W \approx \mathrm{e}^{i 2\pi \frac{p}{q}}$. Then, the modified formula for B can be stated as follows:

$$B = \begin{cases} -p \log 2 + C_1 \log N + C_2, & \text{if } q \geq N, \\ S_3 - \max(S_1, S_2) + C_3 \log N + C_4, & \text{if } q < N, \end{cases} \quad (64)$$

where C_3 and C_4 are implementation-dependent constants. For calculations with both double and quadruple floating-point precision, our experiments indicate that C_3 can be set to -1 and C_4 to 0.

Figure S13 shows the error prediction obtained using Eq. (21) but with the offset term B defined as in Eq. (64). The results show that the modified formula no longer underestimates the numerical error. That is, the points below zero in Fig. S12c are now in line with all other points.

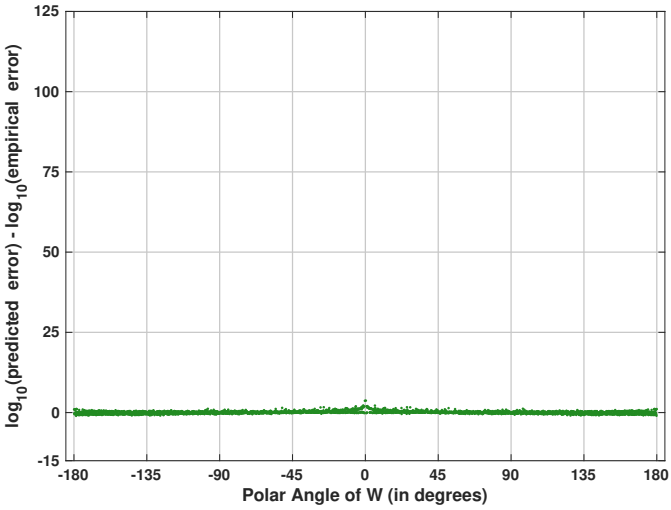

Fig. S13. Visualization of the difference between the logarithms of the predicted and the empirical error, computed using the modified offset term B. This is similar to Fig. S12c, but in this case all predicted errors are close to the empirically-observed errors and all points are plotted in green.