mathematics

MDPI

Article

Numerical Inverse Transformation Methods for Z-Transform

Illés Horváth [1], András Mészáros [2] and Miklós Telek [2,*]

[1] MTA-BME Information Systems Research Group, 1117 Budapest, Hungary; horvath.illes.antal@gmail.com
[2] Department of Networked Systems and Services, Technical University of Budapest, 1117 Budapest, Hungary; meszarosa@hit.bme.hu
* Correspondence: telek@hit.bme.hu; Tel.: +36-1-463-3261

Received: 12 March 2020; Accepted: 3 April 2020; Published: 10 April 2020

Abstract: Numerical inverse Z-transformation (NIZT) methods have been efficiently used in engineering practice for a long time. In this paper, we compare the abilities of the most widely used NIZT methods, and propose a new variant of a classic NIZT method based on contour integral approximation, which is efficient when the point of interest (at which the value of the function is needed) is smaller than the order of the NIZT method. We also introduce a vastly different NIZT method based on concentrated matrix geometric (CMG) distributions that tackles the limitations of many of the classic methods when the point of interest is larger than the order of the NIZT method.

Keywords: inverse Z-transformation; numerical analysis; contour integral; finite order approximation; matrix geometric distribution

1. Introduction

Z-transformation is one of the most frequently used non-linear transformations for describing discrete time series [1]. In several engineering and applied mathematical fields, non-linear transformations provide a compact description of the system behavior. Many practically important operations, e.g., discrete convolution, are much easier to handle in Z-transform domain, which makes the use of Z-transform domain system description widespread in many fields. While a lot of important characteristics (e.g., poles, frequency response, initial/final values, etc.) can be obtained directly from Z-transform domain description, there are plenty of practically important cases where explicit time domain values are needed. In this case, inverse Z-transformation (IZT) must be performed.

In some special cases, symbolic IZT is feasible, but in a wide range of practically important cases numerical IZT (NIZT) is required to compute the time domain values based on the Z-transform domain description. This paper focuses on the problem of NIZT.

Since NIZT has been widely applied in practice for a long time, its literature is rather rich. In this paper, we consider only the most efficient methods of the recent literature [2]. These methods are introduced in the subsequent discussions and are used also for numerical comparison.

Section 2 is devoted to the general setup of Z-transformation. Section 3 gives a brief review of general methods in the literature. From among the few NIZT methods available in the literature, one stands out, described by several authors independently [2–4]. We will refer to this method as the CIR method, and it is described in Section 4 along with an interpretation based on *C*ontour *I*ntegral approximation starting from the positive *R*eal axis. Section 5 provides a variant of this method with the starting point of the *C*ontour *I*ntegral *S*hifted, referred to as the *CIS* method. Section 6 provides an entirely new method, referred to as the *CMG* method, based on *C*oncentrated *M*atrix *G*eometric (CMG)

functions with the necessary background. Section 7 contains numerical comparison of the various NIZT methods. Section 8 concludes the paper.

2. The Z-Transform and Its Inverse

Let $g(t)$ be a series with countably many elements. The (unilateral) Z-transform of $g(t)$ is defined as

$$\mathcal{Z}\{g(t)\} = g^\star(z) = \sum_{\ell=0}^{\infty} z^{-\ell} g(\ell), \tag{1}$$

where $g^\star(z)$ is the Z-transform of $g(t)$.

The inverse transform problem is to find the T-th element of $g(t)$, i.e., $g(T)$ based on $g^\star(z)$. This paper focuses on the case when symbolic inverse Z-transformation is available and NIZT is required to find an approximate value of $g(T)$ based on $g^\star(z)$.

The region of convergence (ROC) for the summation in (1) is always of the form $\{z : |z| > c\}$, possibly including some points of the boundary $|z| = c$. (1) is absolute convergent on $\{z : |z| > c\}$, divergent on $\{z : |z| < c\}$, and on the boundary can be either absolute convergent, convergent or divergent. The region may also be empty ($c = \infty$), or the entire complex plane ($c = 0$). The real constant c is known as the limit of absolute convergence. In case c is finite, the function $\mathcal{Z}\{g(t)\}$ may extend analytically to a domain larger than the region of convergence (e.g., for $g(t) = q^t$, (1) is convergent for $|z| > q$, but $\mathcal{Z}\{g(t)\} = \frac{z}{z-q}$ extends analytically to $\mathbb{C} \setminus \{q\}$).

The inverse at point T can be obtained from the contour integral [1]

$$g(T) = \frac{1}{2\pi\Im} \oint_C g^\star(z) z^{T-1} dz, \tag{2}$$

where C is a counterclockwise closed path encircling the origin and entirely in the region of convergence, and i is the complex unit.

The Z-transform has a natural scaling property:

$$g^\star(az) = \sum_{\ell=0}^{\infty} z^{-\ell} a^{-\ell} g(\ell) = \mathcal{Z}\{a^{-t} g(t)\}, \tag{3}$$

which allows any NIZT method to be applied to $g^\star(az)$ instead of $g^\star(z)$ to approximate $a^{-T} g(T)$ (and then $g(T)$ by multiplying by a^T). In some cases, the scaling with a has special analytical interpretation; we provide such interpretation for contour integral methods. In the numerical section we study the effect of a for all the methods included in the present paper.

In this work, we assume that $g(t)$ is real for any non-negative integer t. This assumption is well-suited for most practical applications and implies $g^\star(\bar{z}) = \bar{g}^\star(z)$ (where $\bar{z}$ denotes the complex conjugate of z). Consequently, it is sufficient to evaluate $g^\star(z)$ only at one of each complex conjugate pair.

3. General Inverse Z-Transformation Methods

In this section, we provide a short overview of various NIZT methods proposed in the literature that do not use the contour integral in (2) for the inverse Z-transformation. The contour integral-based NIZT approach of [2–4] will be discussed separately in Section 4.

3.1. Inverse Transformation Based on Moments

In [5], Tagliani proposes a method for NIZT that requires the availability of a finite number of the transform's derivatives. The derivatives are used to calculate the moments of $g(t)$, and based on these moments an approximating "analytical form" is calculated. Consequently, while the author presents it as an inverse Z-transform method, it is more of a moment fitting algorithm. The benefit of Tagliani's approach is that it can be used as long as the moments of $g(t)$ are obtainable even if only a functional

equation is available for the Z-transform. However, the performance of the method is significantly worse than numerical integration-based methods both in terms of precision and in computation time.

3.2. Inverse Transformation Based on Orthogonal Decomposition

Rajković et al. propose an inversion method in [6] that approximates $g(t)$ with

$$g_{N,q}(t) = \sum_{n=0}^{N} c_n \varphi_q^{(n)}(t),$$

where

$$\varphi_q^{(n)}(t) = \sum_{k=1}^{n} b_{q,n,k} q^{kt}.$$

Parameter q can be chosen freely, but should be slightly smaller than 1 (in the numerical experiments of [6] $q = 3/4$ and $q = 5/6$ are used). The $b_{q,n,k}$ coefficients are calculated as

$$b_{q,n,k} = (-1)^{n-k} q^{-\binom{n}{2}+\binom{k+1}{2}-kn} \begin{bmatrix} n \\ k \end{bmatrix} \begin{bmatrix} n+k-1 \\ k-1 \end{bmatrix},$$

where

$$[n] = \frac{1-q^n}{1-q}, \quad [n]! = [n][n-1]\dots[1], \quad \begin{bmatrix} n \\ k \end{bmatrix} = \frac{[n]!}{[n-k]![k]!}.$$

The c_n coefficients are calculated as

$$c_n = \frac{q^{n(1-n)}}{1-q^{2n}} \sum_{j=1}^{k} b_{q,k,j} g^\star(1/q^j).$$

The above c_n parameters will minimize $||g(t) - g_N(t)||^2$ for the given $\varphi_q^{(1)}(t), \dots, \varphi_q^{(N)}(t)$ set of series. The idea behind the method is that the $b_{q,n,k}$ parameters are chosen such that $\varphi_q^{(1)}(t), \dots, \varphi_q^{(N)}(t)$ is a set of orthogonal series, i.e., $\sum_{t=0}^{\infty} \varphi_q^{(j)}(t) \varphi_q^{(k)}(t) = \delta_{j,k} r(q)\ \forall j,k \in \mathbb{N}$, where $\delta_{j,k}$ is the Kronecker-delta ($\delta_{j,k} = 1$ if $j = k$ and $\delta_{j,k} = 0$ otherwise) and $r(q)$ is a function of q. It is the consequence of this orthogonality that the c_n values calculated as above are optimal (in 2-norm).

3.3. Inverse Transformation Based on a Linear System of Equations

The $g(t)$ series can also be approximated based on a simple truncation of the Z-transform presented by Merrikh–Bayat in [4]. From the definition of the Z-transform we have

$$g^\star(z) = \sum_{t=0}^{\infty} g(t) z^{-t} \approx \sum_{t=0}^{N} g(t) z^{-t} \tag{4}$$

By using this approximation in the $z_1, z_2, \dots, z_m$ set of points chosen from the ROC we obtain

$$\begin{pmatrix} g^\star(z_1) \\ g^\star(z_1) \\ \dots \\ g^\star(z_m) \end{pmatrix} \approx \begin{pmatrix} 1 & z_1^{-1} & z_1^{-2} & \dots & z_1^{-N} \\ 1 & z_2^{-1} & z_2^{-2} & \dots & z_2^{-N} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & z_m^{-1} & z_m^{-2} & \dots & z_m^{-N} \end{pmatrix} \times \begin{pmatrix} g(0) \\ g(1) \\ \dots \\ g(N) \end{pmatrix}.$$

If $m = N$, the above equation has a unique solution (assuming that the matrix is non-singular, which is true if $z_j \neq z_k, \forall j \neq k$). The issue with the $m = N$ case is that it leads to an ill-conditioned

problem, thus Merrikh–Bayat proposes to choose $m \approx 1.1N$ then minimize $||\boldsymbol{A}\underline{g}_t - \underline{g}_z||_2$, where $\underline{g}_t = [g(0), \ldots, g(N)]$, $\underline{g}_z = [g^\star(z_1), \ldots, g^\star(z_m)]$, and

$$\boldsymbol{A} = \begin{pmatrix} 1 & z_1^{-1} & z_1^{-2} & \ldots & z_1^{-N} \\ 1 & z_2^{-1} & z_2^{-2} & \ldots & z_2^{-N} \\ \vdots & \vdots & \vdots & \vdots & \\ 1 & z_m^{-1} & z_m^{-2} & \ldots & z_m^{-N} \end{pmatrix}.$$

Minimization of $||\boldsymbol{A}\underline{g}_t - \underline{g}_z||_2$ is a least squares problem, which is done using QR decomposition or singular value decomposition in [4]. These have a computational cost of $O(m^2N)(= O(N^3)$ when $m \approx 1.1N)$ for an $m \times N$ matrix, thus this method is computationally expensive compared to numerical integration-based methods.

The error in (4) is small when $g(t)$ is rapidly decaying and N is large enough so that $g(t), 0 \leq t < N$ captures most of the significant values in the sequence. Accordingly, numerical experiments show that the method gives best results for rapidly decaying $g(t)$ sequences when N is sufficiently large; however, increasing N further does not seem to improve the error. Overall, the non-vanishing error renders the applicability of this method limited. This is addressed further in Section 7.

4. Contour Integral-Based Inverse Z-Transformation Methods

Equation (2) can be rewritten as

$$g(T) = \frac{1}{2\pi} \int_0^{2\pi} g^\star(ae^{\Im\omega}) \left(ae^{\Im\omega}\right)^T \mathrm{d}\omega, \tag{5}$$

where $\Im$ is the complex unit. Equation (6) corresponds to the case when C is the circle of radius a in (2). a is actually equivalent to the scaling parameter in (3), as shown in Remark 4.

Contour integral-based methods, in general, approximate integral (5) with the finite sum

$$g(T) \approx g_N(T) = \frac{1}{2\pi} \sum_{k=1}^{N} (\omega_k - \omega_{k-1}) g^\star(ae^{\Im\omega_k})(ae^{\Im\omega_k})^T, \tag{6}$$

where $\omega_k, k = 0, 1, 2, \ldots, N$ define a properly chosen partition of $[0, 2\pi]$ and N is the order of the approximation.

This method is described in [3,4,7] in a slightly different manner independently from each other. In theory any ω_k partition could be chosen, but the above papers use

$$\omega_k = \frac{2k\pi}{N}, \qquad k = 1, 2, \ldots, N \tag{7}$$

exclusively. Since the ω_k's are equidistant with $\omega_k - \omega_{k-1} = \frac{2\pi}{N}$, (6) can be further simplified as

$$g_N(T) = \frac{1}{N} \sum_{k=1}^{N} g^\star(ae^{\Im\omega_k})(ae^{\Im\omega_k})^T = \frac{1}{N} \sum_{k=1}^{N} g^\star(a\beta_k)(a\beta_k)^T, \tag{8}$$

where

$$\beta_k = e^{\Im\omega_k} = \exp\left(\frac{2k\pi\Im}{N}\right), \qquad k = 1, 2, \ldots, N. \tag{9}$$

The β_k parameters are referred to as *nodes* in the rest of the paper. Choosing the nodes according to (9) has several consequences:

1. Since the β_ks consist of complex conjugate pairs (along with real numbers 1 (and also (-1) for even N)), approximation (8) is guaranteed to be real.
2. In case one of the $a\beta_k$ values coincides with a pole of $g^\star$, (8) cannot be evaluated (even if they are close, there is numerical instability). A typical example is when $g^\star$ has a pole at 1 and $a = 1$.
3. Selecting a to be larger than the limit of absolute convergence c guarantees that (8) avoids all poles of $g^\star$. That said, selecting a too large may also cause issues: depending on the function g (and $g^\star$) evaluating $g^\star(a\beta_k)$ with sufficient precision might be difficult; also, the large factor a^T may cause numerical instability if there are cancellations in the sum in (8). The choice of a that provides the most accurate estimate for $g(T)$ is highly dependent on g (and $g^\star$) and can be difficult to determine in general.
4. We have

$$g_N(T) = \frac{1}{N}\sum_{k=1}^{N} g^\star(a\beta_k)(a\beta_k)^T = a^T \cdot \frac{1}{N}\sum_{k=1}^{N} g^\star(a\beta_k)\beta_k^T,$$

   showing that in accordance with (3), (8) is indeed the same numerical method applied to $g^\star(az)$ instead of $g^\star(z)$.
5. Due to complex conjugate pairs, the actual number of evaluations of $g^\star$ to compute (8) is $\lfloor (N+1)/2 \rfloor$. However, we stick with N in the notation as N is in general a better indicator for the properties of the approximation.

4.1. FFT-Based Implementation

Calculating $g_N(T)$ for $T = 0, 1, \ldots, N-1$ with a naive approach has $O(N^2)$ computational cost. This can be reduced, however, by realizing that (8) has the form of an inverse discrete Fourier transform (IDFT). The IDFT of the sequence $\{X_0, X_1, \ldots, X_{N-1}\}$ is $\{x_0, \ldots, x_{N-1}\}$ with

$$x_T = \frac{1}{N}\sum_{k=0}^{N-1} X_k \exp\left(\frac{2kT\pi\Im}{N}\right),$$

thus $g_N(0), g_N(1), \ldots, g_N(N-1)$ is the IDFT of $g^\star(a\beta_1), g^\star(a\beta_2), \ldots, g^\star(a\beta_N)$.

The IDFT of the $g_N(0)$, $g_N(1)$, $\ldots$, $g_N(N-1)$ sequence can be calculated using fast Fourier transform (FFT) algorithms, which have $O(N\log(N))$ computational cost. The first FFT algorithm was presented by Cooley and Tukey in [8]. Its most commonly used radix-2 version requires that $N = 2^k, k \in \mathbb{N}$. Efficient implementations of the radix-2 Cooley-Tukey algorithm are available in most major programming languages (see, e.g., [9]). There are multiple other FFT methods (e.g., Rader's method, the prime-factor algorithm [10], Bluestein's algorithm [11], etc.). Some of these can calculate the IDFT in $O(N\log(N))$ time even for prime N; however, in general, choosing $N = 2^k$ is the most efficient.

4.2. Interpretation Using Approximate Dirac function

This section provides a different interpretation of the approximating function $g_N(T)$. Substituting (1) into (8) gives

$$\begin{aligned} g_N(T) &= \frac{1}{N}\sum_{k=1}^{N} g^\star(a\beta_k)(a\beta_k)^T = \frac{1}{N}\sum_{k=1}^{N}\sum_{\ell=0}^{\infty}(a\beta_k)^{-\ell} g(\ell)(a\beta_k)^T \\ &= \sum_{\ell=0}^{\infty} g(\ell)\frac{1}{N}\sum_{k=1}^{N}(a\beta_k)^{T-\ell} = \sum_{\ell=0}^{\infty} g(\ell) f_{N,a}(T-\ell), \end{aligned} \tag{10}$$

where

$$f_{N,a}(\ell) = \frac{1}{N}\sum_{k=1}^{N}(a\beta_k)^{\ell} = \frac{1}{N}\sum_{k=1}^{N} a^{\ell}\exp\left(\frac{2k\ell\pi\Im}{N}\right). \tag{11}$$

(10) tells us that $g_N(t)$ is a convolution of the original $g(\ell)$ with the function $f_{N,a}(\ell)$; g_N will approximate g well if $f_{N,a}(\ell)$ is close to the function δ_0 (which is defined to be 1 at 0 and 0 at every other integer).

At integer points $f_{N,a}(\ell)$ simplifies to

$$f_{N,a}(\ell) = \begin{cases} a^{\ell}, & \text{for } \ell = K\cdot N, K\in\mathbb{Z}, \\ 0, & \text{otherwise,} \end{cases} \tag{12}$$

thus $f_{N,a}(0) = 1$, and the closest non-zero values are at $\pm N$. As $N\to\infty$, $f_{N,a}(\ell)$ converges to δ_0 at integer points.

(10) and (12) also ensure that $g_N(T)$ is N-periodic except for the exponential factor introduced by a, i.e.,

$$g_N(N+T) = a^{-N}g_N(T). \tag{13}$$

This periodic behavior indicates that for non-periodic $g(t)$ functions the $g_N(T)$ approximation might be poor.

The error of the approximation is

$$\begin{aligned} g_N(T) - g(T) &= \sum_{\ell=0}^{\infty} f_{N,a}(T-\ell)g(\ell) - g(T) \\ &= \sum_{\ell=0}^{T-1} f_{N,a}(T-\ell)g(\ell) + \sum_{\ell=T+1}^{\infty} f_{N,a}(T-\ell)g(\ell) \\ &= \sum_{\ell=1}^{T} f_{N,a}(\ell)g(T-\ell) + \sum_{\ell=1}^{\infty} f_{N,a}(-\ell)g(T+\ell). \end{aligned} \tag{14}$$

Remarks about (14):

1. As long as $N > T$, the first sum vanishes, since $f_{N,a}(T) = 0$ for $T = 1, 2, \ldots, N-1$.
2. Based on (12), (14) can be written as

$$\begin{aligned} g_N(T) - g(T) = \sum_{k\geq -\lfloor T/N\rfloor, k\neq 0} a^{-kN}g(T+kN) = \\ \sum_{k=1}^{\lfloor T/N\rfloor} a^{kN}g(T-kN) + \sum_{k=1}^{\infty} a^{-kN}g(T+kN). \end{aligned} \tag{15}$$

   (15) can be intuitively understood as cutting the sequence $g(T)$ into sections of length N, shifting them over the same interval and summing them, see Figure 1. The first section ($K = 0$) corresponds to values of the original function over the interval $0 \leq T < N$, while the rest corresponds to the error. One consequence of (15) is that for non-decaying or slowly decaying $z(T)$ sequences, selecting $a > 1$ is necessary to ensure fast decay of the error.
3. If $a > 1$, then the first sum in (15) is magnified and the second sum is diminished. This is particularly useful when $N > T$, since in that case the first sum vanishes and the second sum can be diminished arbitrarily (at the cost of loss of precision, more on this later).
4. If $a < 1$, the first sum is diminished, and the second sum is magnified.

5. If $N \leq T$, the first sum in (15) does not vanish, and is magnified when choosing $a < 1$. However, choosing $a > 1$ magnifies the second sum in (15) instead; altogether, this results in a non-vanishing error regardless of the choice of a. Consequently, the classic method (or any contour integral-based method) has significant approximation error when $N \leq T$, as shown by the numerical results in Section 7.
6. When $g(\ell) > 0$, according to (12) the second sum has positive terms only, so there are no cancellations. On the other hand, the approximation preserves nonnegativity, which might be a relevant property in certain applications.
7. If $g(\ell)$ has positive and negative values alternating, then there can be cancellations according to (15) which reduce the corresponding error.

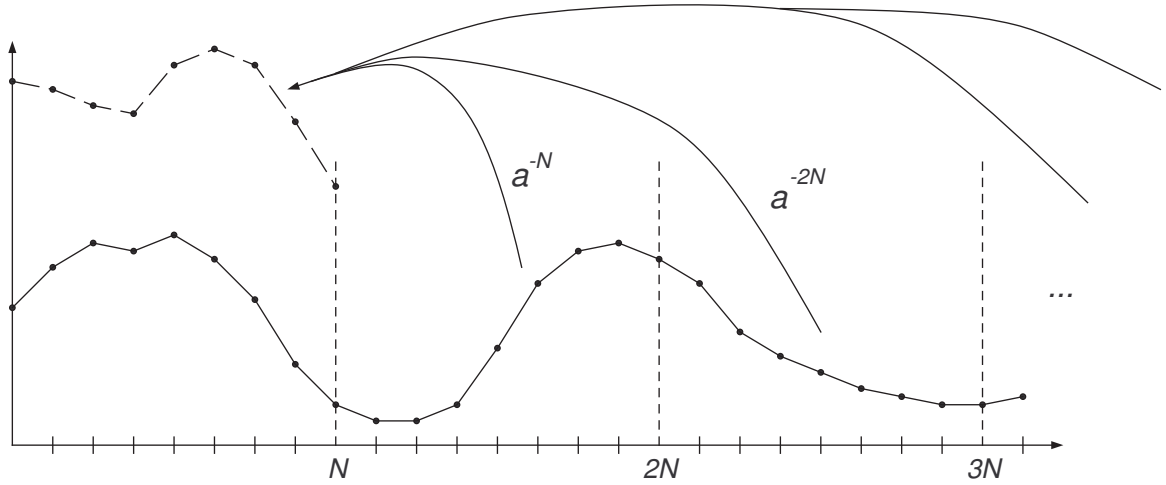

Figure 1. Sections shifted and summed according to (15) when $T < N$.

5. Shifting the Nodes of the CIR Method

In this section, we present a variant of the CIR method, referred to as CIS method, where the nodes are shifted by a half period compared to (7), i.e., we propose to use

$$\beta_k^{(2)} = \exp\left(\frac{(2k-1)\pi\Im}{N}\right), \qquad k = 1, 2, \ldots, N \tag{16}$$

in (6) instead of (7). Figures 2 and 3 display the positioning of the β_k nodes for the CIR and the $\beta_k^{(2)}$ nodes for the CIS methods for even and odd values of N.

Theorem 1. *Applying the $\beta_k^{(2)}$ nodes partition in the contour integral-based NIZT according to (8), results in the NIZT procedure*

$$g_N^{(2)}(T) = \frac{1}{N}\sum_{k=1}^{N} g^{\star}(a\beta_k^{(2)})(a\beta_k^{(2)})^T, \tag{17}$$

whose error is

$$g_N^{(2)}(T) - g(T) = \sum_{\ell=1}^{T} f_{N,a}^{(2)}(\ell)g(T-\ell) + \sum_{\ell=1}^{\infty} f_{N,a}^{(2)}(-\ell)g(T+\ell), \tag{18}$$

where

$$f_{N,a}^{(2)}(\ell) = \begin{cases} (-1)^K a^\ell, & \text{for } \ell = K \cdot N, K \in \mathbb{Z}, \\ 0, & \text{for } \ell \in \mathbb{N} \setminus \{K \cdot N : K \in \mathbb{Z}\}. \end{cases} \tag{19}$$

As $N \to \infty$, $f_{N,a}^{(2)}(\ell)$ *converges to* δ_0 *at integer points.*

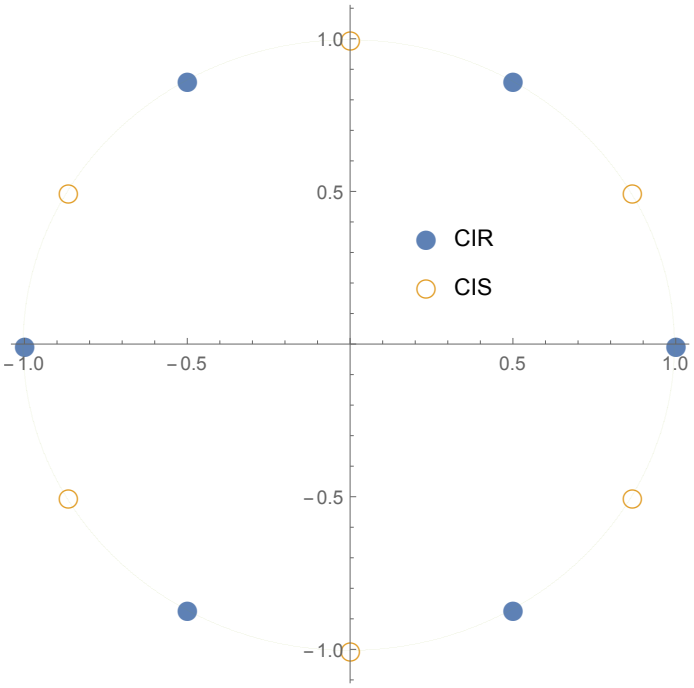

Figure 2. Nodes of the CIR and CIS NIZT methods for $N = 6$ in the complex plane.

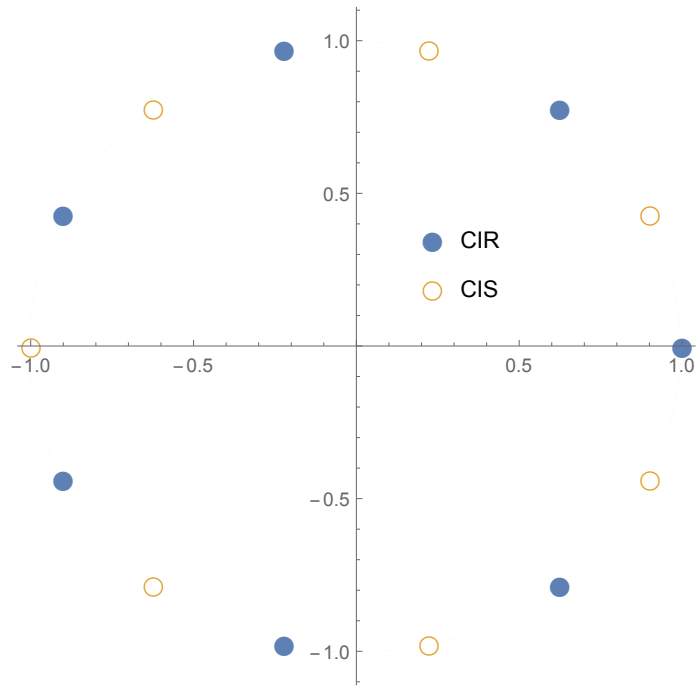

Figure 3. Nodes of the CIR and CIS NIZT methods for $N = 7$ in the complex plane.

Proof. Applying the $\beta_k^{(2)}$ values in (8) gives (17).

Similar to (10), we have

$$g_N^{(2)}(T) = \frac{1}{N}\sum_{k=1}^{N} g^\star(a\beta_k^{(2)})(a\beta_k^{(2)})^T = \frac{1}{N}\sum_{k=1}^{N}\sum_{\ell=0}^{\infty}(a\beta_k^{(2)})^{-\ell} g(\ell)(a\beta_k^{(2)})^T$$
$$= \sum_{\ell=0}^{\infty} g(\ell)\frac{1}{N}\sum_{k=1}^{N}(a\beta_k^{(2)})^{T-\ell} = \sum_{\ell=0}^{\infty} g(\ell) f_{N,a}^{(2)}(T-\ell), \tag{20}$$

where

$$f_{N,a}^{(2)}(\ell) = \frac{1}{N}\sum_{k=1}^{N}(a\beta_k^{(2)})^{\ell}, \tag{21}$$

which satisfies (19) in integer points. Subtracting $g(T)$ from both sides of (20) gives (18). □

Remarks about (18):

1. $f_{N,a}(0) = 1$, and the closest non-zero values are at $\pm N$, which are negative.
2. As long as $N > T$, the first sum vanishes, since $f_{N,a}^{(2)}(T) = 0$ for $T = 1, 2, \ldots, N-1$.
3. According to (19), (18) can be rewritten as

$$g_N^{(2)}(T) - g(T) = \sum_{k\geq -\lfloor T/N\rfloor, k\neq 0} (-a)^{-kN} g(T+kN) =$$
$$= \sum_{k=1}^{\lfloor T/N\rfloor} (-a)^{kN} g(T-kN) + \sum_{k=1}^{\infty}(-a)^{-kN} g(T+kN). \tag{22}$$

4. If $a > 1$, then the first sum is magnified, and the second sum is diminished. This is particularly useful when $N > T$, since in that case the first sum vanishes, and the second sum can be diminished arbitrarily.
5. If $a < 1$, the first sum is diminished, and the second sum is magnified.
6. When $g(T) > 0$, (22) has alternating terms, so there are cancellations in the second sum, reducing the corresponding error. This is particularly useful when $g^\star$ has a pole at c, since the $\beta_k^{(2)}$ values do not include a after the node shift, so (17) can still be evaluated with $a = c$, and the error from the second sum will be smaller due to cancellations. On the other hand, unlike for CIR, the approximation does not necessarily preserve nonnegativity.
7. If $g(T)$ is alternating, then there are no cancellations in (22).
8. Due to complex conjugate pairs, the actual number of evaluations of $g^\star$ necessary to compute (17) is $\lfloor N/2\rfloor$. However, just like for CIR, we stick with N as the notation.

6. Concentrated Matrix Geometric Distribution-Based Inverse Transformation

As we have seen in Remark 5 of (14), contour integral-based approximation methods are generally ill-suited to approximate $g(T)$ when the node number is smaller than T, i.e., $N \leq T$. If increasing N further is not feasible (e.g., because the computational cost of evaluating $g^\star(z)$ N times gets too large, or due to the precision loss of the procedure with the given floating point number representation), then a different NIZT procedure is needed.

In this section, we propose an approach for NIZT based on concentrated matrix geometric distributions, which we thus call CMG method and is the application of the CME (concentrated matrix exponential) method [12] for discrete time. The CME method is a numerical inverse Laplace transformation (NILT) procedure that uses the Abate–Whitt framework [13]. In the following, we first introduce the Abate–Whitt framework, then we present the CME method, finally we show how it can be applied to discrete time to obtain the proposed CMG method.

6.1. Abate–Whitt Framework for Numerical Inverse Laplace Transformation

The Laplace-transform of a function $h(t)$ is defined as

$$h^*(s) = \int_{t=0}^{\infty} e^{-st} h(t) \mathrm{d}t.$$

The inverse transform problem is to find an approximate value of function h at point T (i.e., $h(T)$) based on $h^*(s)$. The Abate–Whitt framework uses the following form for this approximation:

$$h(T) \approx h_n(T) := \sum_{k=1}^{n} \frac{\eta_k}{T} h^* \left(\frac{\beta_k}{T} \right), \quad T > 0.$$

This approximation has a simple interpretation based on the reformulation

$$h_n(T) = \frac{1}{T} \sum_{k=1}^{n} \eta_k h^* \left(\frac{\beta_k}{T} \right) = \int_0^{\infty} h(t) \cdot \frac{1}{T} f_n \left(\frac{t}{T} \right) \mathrm{d}t, \tag{23}$$

where

$$f_n(t) = \sum_{k=1}^{n} \eta_k e^{-\beta_k t}. \tag{24}$$

If $f_n(t)$ was the Dirac impulse function at point T, then the Laplace inversion would be perfect, but depending on the weights η_k and nodes β_k, the function $f_n(t)$ only approximates the Dirac impulse function with a given accuracy. There are multiple different types of $f_n(t)$ functions that can be used for the approximation [12].

6.2. CME Method

In the CME method the probability density function (pdf) of a matrix exponential distribution is chosen as the $f_n(t)$ function. The class of matrix exponential (ME) distributions of order N contains positive random variables with pdf of the form

$$f_X(t) = -\underline{\alpha} \boldsymbol{A} e^{\boldsymbol{A}t} \underline{\mathbf{1}}, \qquad t \geq 0, \tag{25}$$

where $\underline{\alpha}$ is a real row vector of length N, $\boldsymbol{A}$ is a real matrix of size $N \times N$, and $\underline{\mathbf{1}}$ is a column vector of ones of size N [14]. To ensure $\int_0^{\infty} f_X(t) \mathrm{d}t = 1$, $\underline{\alpha}$ and $\boldsymbol{A}$ are such that $\underline{\alpha}\underline{\mathbf{1}} = 1$ and the eigenvalues of $\boldsymbol{A}$ have negative real part.

Nonnegativity of $f_X(t)$ does not follow from (25), but some $(\underline{\alpha}, \boldsymbol{A})$ pairs result in $f_X(t)$ functions that are non-negative for $t \geq 0$ [15]. When $\boldsymbol{A}$ is diagonalizable with spectral decomposition $\boldsymbol{A} = \sum_{k=1}^{N} u_k \lambda_k v_k$, where λ_k are the eigenvalues, u_k are the right eigenvectors and v_k are the left eigenvectors of $\boldsymbol{A}$ for $k = 1, \ldots N$, then f_X can be written as

$$f_X(t) = \sum_{k=1}^{N} \underbrace{-\underline{\alpha} \boldsymbol{A} \underline{u_k} \, \underline{v_k} \underline{\mathbf{1}}}_{c_k} e^{\lambda_k t} = \sum_{k=1}^{N} c_k e^{\lambda_k t} = \sum_{k=1}^{n} \eta_k e^{-\beta_k t}, \tag{26}$$

with $\eta_k = c_k$ and $\beta_k = -\lambda_k$. Comparing (26) and (24) shows that ME distributions with diagonalizable matrix $\boldsymbol{A}$ can be used in the place of $f_n(t)$ to obtain an ILT method of the Abate–Whitt framework.

As mentioned before, the primary task when using the Abate–Whitt framework is to approximate the Dirac impulse with the $f_n(t)$ function as closely as possible. The squared coefficient of variation (SCV) measures how concentrated a non-negative normalized function on $\mathbb{R}^+$ is, and it is a good indicator of the quality of the approximation. The SCV of f can be calculated as

$$\mathrm{SCV}(f) := \frac{\int_{t=0}^{\infty} t^2 f(t)\mathrm{d}t}{\left(\int_{t=0}^{\infty} t f(t)\mathrm{d}t\right)^2} - 1.$$

Function f with $\mathrm{SCV}(f) = 0$ is the Dirac function and the smaller $\mathrm{SCV}(f)$ is, the better f approximates the Dirac function. The parameters of CME distributions with low SCV have been calculated for up to order 1000 [15] and can be accessed at [16].

The CME method has several advantages compared to other NILT methods [12]. It is more stable numerically, provides smooth, over- and under-shooting free approximation even for discontinuous functions and, contrary to other methods of the family, its precision gradually improves when increasing its order (N). The application of the CME method for NIZT is discussed in the next section.

6.3. CMG Method

A discrete counterpart of the CME method is formulated in the following theorem.

Theorem 2. *For a discrete function $g : \mathbb{Z} \to \mathbb{R}$, defining the continuous function*

$$\hat{g}(t) = \begin{cases} 0, & \text{for } t < 1/2, \\ g(\ell), & \text{for } \ell - 1/2 \le t < \ell + 1/2, \end{cases}$$

and applying the CME inverse Laplace transformation method at point T with weights η_k and nodes β_k, results in the NIZT procedure

$$g_N(T) = \begin{cases} g^\star(0), & \text{for } T = 0, \\ \sum_{k=1}^{N} \bar{\eta}_k \left(g^\star\left(e^{\frac{\beta_k}{T}}\right) - g^\star(0) \right), & \text{for } T > 0, \end{cases} \tag{27}$$

where

$$\bar{\eta}_k = \frac{\eta_k}{\beta_k}\left(e^{\frac{\beta_k}{2T}} - e^{-\frac{\beta_k}{2T}}\right).$$

Proof. The inverse Laplace transformation of $\hat{g}(t)$ according to (23) can be written as

$$\begin{aligned} \hat{g}_N(T) &= \frac{1}{T}\sum_{k=1}^{N} \eta_k \int_{t=0}^{\infty} e^{-\frac{\beta_k}{T}t}\hat{g}(t)\mathrm{d}t = \sum_{\ell=1}^{\infty}\int_{t=\ell-1/2}^{\ell+1/2} g(\ell)e^{-\frac{\beta_k}{T}t}\mathrm{d}t \\ &= \frac{1}{T}\sum_{k=1}^{N}\eta_k \sum_{\ell=1}^{\infty} -\frac{T}{\beta_k}\left(e^{-\frac{\beta_k}{T}(\ell+1/2)} - e^{-\frac{\beta_k}{T}(\ell-1/2)}\right) g(\ell) \\ &= \frac{1}{T}\sum_{k=1}^{N}\eta_k \sum_{\ell=1}^{\infty} \frac{T}{\beta_k}\left(e^{\frac{\beta_k}{2T}} - e^{-\frac{\beta_k}{2T}}\right) g(\ell)e^{-\frac{\beta_k}{T}\ell} \\ &= \sum_{k=1}^{N}\frac{\eta_k}{\beta_k}\left(e^{\frac{\beta_k}{2T}} - e^{-\frac{\beta_k}{2T}}\right)\sum_{\ell=1}^{\infty}\left(e^{\frac{\beta_k}{T}}\right)^{-\ell} g(\ell). \end{aligned}$$

Based on this last expression, we can express $\hat{g}_N(T)$ from $g^\star(z)$ using its definition in (1) as

$$\hat{g}_N(T) = \sum_{k=1}^{N}\frac{\eta_k}{\beta_k}\left(e^{\frac{\beta_k}{2T}} - e^{-\frac{\beta_k}{2T}}\right)\sum_{\ell=1}^{\infty}\left(e^{\frac{\beta_k}{T}}\right)^{-\ell} g(\ell) = \sum_{k=1}^{N}\bar{\eta}_k\left(g^\star\left(e^{\frac{\beta_k}{T}}\right) - g(0)\right) \tag{28}$$

From the definition of the Z-transform we have $g(0) = g^\star(0)$, thus (28) takes the form of (27). □

The scaled version of (27) according to (3) is

$$g_N(T) = \begin{cases} g^\star(0), & \text{for } T = 0, \\ a^T \sum_{k=1}^{N} \bar{\eta}_k \left(g^\star \left(a e^{\frac{\beta_k}{T}} \right) - g^\star(0) \right), & \text{for } T > 0. \end{cases} \tag{29}$$

In general, the interpretation of the CMG method is similar to the one of the CME method for NILT in that the approximation is essentially based on a discrete approximation of the Dirac function. The main advantage of this method compared to many other NIZT methods discussed above is that the error of CMG method remains small also when $T \geq N$.

The parameter a affects the "shape" of the discrete approximation of the Dirac function, such that for $a > 1$, the contribution of the $g(t)$ terms to the error of $g_N(T)$ is magnified for $t < T$ and diminished for $t > T$, while for $a < 1$, it is the other way around. The effect of a on the accuracy of $g_N(T)$ is examined in Section 7.

7. Numerical Examples

To evaluate the numerical properties of the above listed NIZT methods we use a collection of test functions according to Table 1. This set of functions exhibits a wide range of behaviors. For each function, Table 1 lists the name, the t domain form, $g(t)$, the z domain form, $g^\star(z)$, and the limit of absolute convergence, c.

We check the behavior of the NIZT methods by evaluating the difference of the values computed from $g(T)$ with the ones computed from $g^\star(z)$ ($g_N(T)$) with different choices of the order, the largest evaluated point and the scaling factor (N, $T_{\max}$, and a). Apart from the error in the approximation, we also keep track of the precision loss (number of digits lost due to round-off error) during the calculation of $g_N(T)$. The computations were carried out using Wolfram Mathematica. The precision of the applied arithmetic was high enough (200 digits) to dominate the precision loss.

Table 1. Set of test functions.

Function	$g(t)$	$g^\star(z)$	c
Dirac(10)	δ_{10}	z^{-10}	0
Poisson(1)	$\frac{1}{t!}e^{-1}$	$e^{\left(\frac{1}{z}-1\right)}$	0
Heaviside step	1	$\frac{z}{z-1}$	1
Geometric(1/2)	$(1/2)^t$	$\frac{z}{z-1/2}$	1/2
Geometric(−1/2)	$(-1/2)^t$	$\frac{z}{z+1/2}$	1/2
Triangle wave	$\frac{1+(-1)^t}{2}$	$\frac{z^2}{z^2-1}$	1
Polynomial(1/t)	$\frac{1}{t}\mathbf{1}(t \geq 1)$	$-\log(1-z^{-1})$	1
Polynomial(t)	t	$\frac{z}{(-1+z)^2}$	1
Uniform(5, 10)	$\mathbf{1}(5 \leq t \leq 10)$	$\frac{z^6-1}{z^{11}-z^{10}}$	0

7.1. Numerical Properties When $T < N$

When we are interested in the cases with $T < N$, we use the following error measure based on infinity norm over an interval $[0, T_{\max}]$:

$$\|g - g_N\| = \|g - g_N\|_\infty = \max_{0 \leq T \leq T_{\max}} |g(T) - g_N(T)| \tag{30}$$

$T_{\max}$ is the largest point we are interested in, i.e., we assume $T \leq T_{\max} < N$ in this subsection.

7.1.1. General Comparison of the NIZT Methods

For this comparison, we included the NIZT methods from the previous sections which provide the best results. Ort1 and Ort2 refer to the method presented in Section 3.2 and [6] (using the suggested $q = 3/4$ and $q = 5/6$ parameters), based on orthogonal functions. MB refers to the method in Section 3.3, based on matrix pseudoinverse calculation.

Table 2 compares the error of all methods for the list of functions in Table 1 for $N = 64$ and $T_{\max} = 32$. In the table,"$\sim$0" means practical zero, a value smaller than 10^{-100}, "p.inf." stands for practical infinity, denoting errors larger than 10^2, while "n/a" means not applicable due to a pole of $g^\star$ which is evaluated by the given NIZT method (e.g., the CIR method with $a = 1$ fails for functions with a pole at 1). All calculations related to Table 2 were carried out using high precision (200 digits) floating point arithmetic.

According to Remark 4 at the end of Section 4 (and Remark 5 at the end of Section 5), as long as $T_{\max} < N$, the accuracy of the contour integral-based methods CIR and CIS improves as a is increased, and this also helps avoiding the pole at 1 for the CIR method. Table 3 displays this effect by setting $a = 2$ instead of $a = 1$.

Based on Tables 2 and 3, we conclude that the Ort1 method gives the most precise results, the CIR, CIS give precise results when a is sufficiently large, the Ort2 and MB methods are unreliable, and CMG is relatively reliable for $a = 1$ (although the error is not as small as for CIR, CIS and Ort1), but unreliable for $a = 2$. Altogether, one might have the impression that Ort1 is the best method; however, we have not yet examined other important questions like precision loss or running time. In the next subsection, precision loss is examined along with a more detailed analysis of the role of a.

Due to their unreliability (c.f. Tables 2 and 3) methods Ort2 and MB are excluded from further investigations.

Table 2. Errors for various test functions ($N = 64$, $T_{\max} = 32, a = 1$).

Function	CIR	CIS	Ort1	Ort2	MB	CMG
Dirac(10)	$\sim$0	$\sim$0	$\sim$0	p.inf.	$\sim$0	1.77×10^{-2}
Poisson(1)	2.90×10^{-90}	2.90×10^{-90}	$\sim$0	p.inf.	7.58×10^{-30}	3.17×10^{-5}
Heaviside step	n/a	0.500	5.10×10^{-42}	p.inf.	p.inf.	$\times10^{1.82-5}$
Geometric(1/2)	5.42×10^{-20}	5.42×10^{-20}	7.72×10^{-66}	p.inf.	p.inf.	3.82×10^{-5}
Geometric($-1/2$)	5.42×10^{-20}	5.42×10^{-20}	4.45×10^{-59}	p.inf.	1.04×10^{-7}	3.82×10^{-5}
Triangle wave	n/a	0.500	5.11×10^{-42}	p.inf.	p.inf.	0.431
Polynomial($1/t$)	n/a	1.08×10^{-2}	5.69×10^{-44}	p.inf.	p.inf.	6.54×10^{-5}
Polynomial(t)	n/a	31.5	4.56×10^{-40}	p.inf.	p.inf.	3.35×10^{-3}
Uniform(5,10)	$\sim$0	$\sim$0	$\sim$0	p.inf.	0.181	1.40×10^{-2}

Table 3. Errors for various test functions ($N = 64$, $T_{\max} = 32, a = 2$).

Function	CIR	CIS	Ort1	Ort2	MB	CMG
Dirac(10)	$\sim$0	$\sim$0	$\sim$0	p.inf.	$\sim$0	p.inf.
Poisson(1)	$\sim$0	$\sim$0	$\sim$0	p.inf.	$\sim$0	p.inf.
Heaviside step	5.42×10^{-20}	5.42×10^{-20}	1.66×10^{-56}	p.inf.	p.inf.	p.inf.
Geometric(1/2)	2.94×10^{-39}	2.94×10^{-39}	$\sim$0	p.inf.	p.inf.	p.inf.
Geometric($-1/2$)	2.94×10^{-39}	2.94×10^{-39}	$\sim$0	p.inf.	7.17×10^{-8}	p.inf.
Triangle wave	5.42×10^{-20}	5.42×10^{-20}	4.78×10^{-50}	p.inf.	p.inf.	p.inf.
Polynomial($1/t$)	8.47×10^{-22}	8.47×10^{-22}	1.48×10^{-58}	p.inf.	p.inf.	p.inf.
Polynomial(t)	5.15×10^{-18}	5.15×10^{-18}	1.53×10^{-54}	p.inf.	p.inf.	p.inf.
Uniform(5,10)	$\sim$0	$\sim$0	$\sim$0	p.inf.	$\sim$0	p.inf.

7.1.2. Precision Loss and the Effect of a

For a more detailed view on the performance of the methods CIR, CIS, Ort1, and CMG, we investigate their accuracy as a function of parameter a.

Table 4 contains the error and precision loss (p.l., in digits, calculated using Wolfram Mathematica) for the Polynomial($1/t$) function with $N = 64$, $T_{\max} = 32$, and a taking the values $1/2, 1, 11/10, 2, 4$.

For contour integral methods CIR and CIS, as long as $N > T$, setting a to a larger value will diminish the error in the second term of (14) and (18), while the first term cancels out entirely.

Table 4. The error $\|g - g_N\|$ and precision loss for Polynomial($1/t$) function, $N = 64$, $T_{\max} = 32$.

	CIR		**CIS**		**Ort1**		**CMG**	
a	**Error**	**p.l.**	**Error**	**p.l.**	**Error**	**p.l.**	**Error**	**p.l.**
$1/2$	0.693	12	0.693	3	1.84×10^{-29}	185	7.52×10^{-2}	1
1	n/a	n/a	1.08×10^{-2}	4	5.69×10^{-44}	162	6.54×10^{-5}	1
$11/10$	0.214	5	0.212	5	1.59×10^{-42}	162	2.33×10^{-2}	1
2	5.15×10^{-18}	20	5.15×10^{-18}	20	1.53×10^{-54}	143	p.inf.	1
4	2.79×10^{-37}	38	2.79×10^{-37}	38	$\sim$0	142	p.inf.	1

For $a = 1/2$, the approximations in Table 4 are poor. This is because for $a < 1$, the error in the tail is magnified. If $g(t)$ is rapidly decaying, then the error introduced by $a = 0.5$ is small, but $a = 1$ or $a > 1$ are still better choices.

For $a = 1$, the "n/a" values for the CIR method are due to the pole of the function at 1. This is where the CIS method has an advantage: the shifted nodes avoid the pole when $a = 1$, so it gives a meaningful result.

For $a > 1$, the error decreases rapidly for the contour integral methods CIR and CIS as the error in the tail is diminished, at the cost of increased precision loss. Interestingly, the precision loss is of similar order as the error. (This is not necessarily the case in general, but the precision loss does seem to increase rapidly with a in general.)

The error for the CIS method is slightly smaller than for the CIR due to cancellations (see Remark 6 after (18)), but the difference is practically negligible.

The Ort1 method, while gives the lowest error, suffers from huge precision losses. Notably, for any calculations with precision smaller than 142 digits, Ort1 would give meaningless results due to precision loss. The precision loss is inherent to the Ort1 method due to the highly fluctuating orthogonal functions involved. Overall, we recommend avoiding the use of the Ort1 method due to its unpredictable high precision loss, and instead we recommend using either CIR or CIS when $T < N$, with a set to as large as possible, depending on the precision loss tolerated.

Interestingly, with the Polynomial($1/t$) function the CMG method works best when setting $a = 1$. An intuitive explanation of this property is as follows. For the CMG method, $a > 1$ enlarges the errors that are caused by the non-zero $g(t)$ values for $t < T$ and diminishes errors that are caused by the non-zero $g(t)$ values for $t > T$, and $a < 1$ has the opposite effect. Since Polynomial($1/t$) is a rather flat function the effect of enlarging the error is more dominant for both $a > 1$ and $a < 1$. This property might slightly change for steeper functions, but, in general, we recommend using $a = 1$ for the CMG method. We also note that the CMG method involves practically no loss of precision, so it can be used efficiently even with standard precision floating point arithmetic.

Table 5 investigates the error and the precision loss for various Z-transform $g^\star$ functions using the CIR, CIS and Ort1 methods with parameters $N = 64$, $T_{\max} = 32$, $a = 2$. We note that the CMG method fails due to $a > 1$.

Table 5. Error and precision loss for various test functions ($N = 64, T_{\max} = 32, a = 2$).

	CIR		CIS		Ort1		CMG	
	Error	**p.l.**	**Error**	**p.l.**	**Error**	**p.l.**	**Error**	**p.l.**
Dirac(10)	~ 0	19	~ 0	19	~ 0	92	p.inf.	1
Poisson(1)	~ 0	107	~ 0	107	~ 0	174	p.inf.	1
Heaviside step	5.42×10^{-20}	21	5.42×10^{-20}	21	1.66×10^{-56}	141	p.inf.	1
Geom(1/2)	2.94×10^{-39}	39	2.94×10^{-39}	39	~ 0	149	p.inf.	1
Geom(−1/2)	2.94×10^{-39}	39	2.94×10^{-39}	39	~ 0	149	p.inf.	1
Triangle wave	5.42×10^{-20}	21	5.42×10^{-20}	21	4.78×10^{-50}	199	p.inf.	1
Polynomial($1/t$)	8.47×10^{-22}	22	8.47×10^{-22}	22	1.48×10^{-58}	139	p.inf.	1
Polynomial(t)	5.15×10^{-18}	20	5.15×10^{-18}	20	1.53×10^{-54}	143	p.inf.	1
Uniform(5,10)	~ 0	20	~ 0	20	~ 0	115	116	1

From Table 5, we conclude that the precision loss heavily depends on the $g^\star$ function. As a result, high precision calculations with precision loss check is recommended for NIZT with the CIR, CIS and Ort1 methods. However, with high precision calculations, the precision loss remains tolerable for CIR and CIS.

7.2. Numerical Properties of NIZT Methods When $T \geq N$

Finally, we examine the case when $T \geq N$. This case is relevant when sampling of the Z-transform $g^\star$ is costly, but we still need to approximate $g(T)$ for large values of T.

7.2.1. Failure of Contour Integral Methods

To start off, we display the remarks at the end of Section 4.2 in practice. Table 6 shows the result of applying the CIR approximation method to the z-transform of the Poisson(1) distribution with order $N = 4$ and various choices of a, compared with the actual values of the Poisson distribution.

For $a = 1$, the approximation is relatively accurate up to $T = N - 1 = 3$, but there is a sharp change in the behavior at $T = N$: the approximation is periodic with a period of N, rendering the approximation values for $T \geq N$ useless (see also Figure 1). For $a = 2$, the approximation is even more accurate up to $T = N - 1$, but the error is magnified for $T \geq N$ (due to the scaling by a^N in accordance with (13), e.g., $g_N(4) = 5.9014 = a^4 \cdot g_N(0) = 16 \cdot 0.3688$ in this case). For $a = 1/2$, the error for the $T \geq N$ terms is diminished, but the approximation up to $T = N - 1$ is much less accurate. Altogether, the CIR method cannot be used to obtain an approximation suitable for both $T < N$ and $T \geq N$. The CIS method suffers from the same issues.

Table 6. Order 4 CIR method approximation of the Poisson(1) distribution.

T	**0**	**1**	**2**	**3**	**4**	**5**	**6**
$a = 1/2$	0.6155	0.4172	0.1921	0.0625	0.0385	0.0261	0.0120
$a = 1$	0.3832	0.3709	0.1845	0.0614	0.3832	0.3709	0.1845
$a = 2$	0.3688	0.3681	0.1840	0.0613	5.9014	5.8891	2.9436
exact	0.3679	0.3679	0.1839	0.0613	0.0153	0.0031	0.0005

7.2.2. Comparing The Error of NIZT Methods When $T \geq N$

Since the error of the cases when $T < N$ has been considered in Section 7.1, in this section we define the error as

$$\|g - g_N\| = \|g - g_N\|_\infty = \max_{N \leq T < T_{\max}} |g(T) - g_N(T)|. \tag{31}$$

Table 7 contains the error of each method for various functions. The parameters are $T_{\max} = 32$, $N = 16$ and a either 1 or 1.1. In this table, "p.inf." marks elements larger than 10^3. We note that the MB method is not applicable with these parameters, since the pseudoinverse calculation is only applicable when $T_{\max} < 1.1N$.

Based on Table 7, we conclude that as long as $T \geq N$, the error of the CIR and CIS methods ((14) and (18)) is large either in first or the second term depending on whether $a < 1$ or $a > 1$, and the Ort1 and Ort2 methods are also unreliable. Only the CMG method gives meaningful results for the case when $T \geq N$. As for the value of a, we recommend using 1 as generally that seems to give a reliably low error.

Table 7. Errors for various test functions ($N = 16$, $T_{\max} = 32$).

Function	a	CIR	CIS	Ort1	Ort2	CMG
Dirac(10)	1/2	1.53×10^{-5}	1.53×10^{-5}	6.62×10^{-4}	4.77×10^{-3}	2.39×10^{-5}
Dirac(10)	1	1.00	1.00	7.15×10^{-2}	0.305	1.53×10^{-3}
Dirac(10)	11/10	4.59	4.59	0.229	0.541	2.71×10^{-3}
Poisson(1)	1/2	5.61×10^{-6}	5.61×10^{-6}	9.46×10^{-15}	0.312.	4.21×10^{-9}
Poisson(1)	1	0.368	0.368	2.56×10^{-11}	p.inf.	1.04×10^{-4}
Poisson(1)	11/10	1.69	1.69	2.39×10^{-11}	p.inf.	1.00×10^{-3}
Heaviside step	1/2	1.00	1.00	p.inf.	3.58	1.00
Heaviside step	1	n/a	1.50	0.407	p.inf.	9.46×10^{-5}
Heaviside step	11/10	4.87	4.77	8.63×10^{-2}	p.inf.	4.66×10^{-2}
Geometric(1/2)	1/2	n/a	2.29×10^{-5}	3.70×10^{-6}	0.423	1.44×10^{-9}
Geometric(1/2)	1	1.00	1.00	1.93×10^{-10}	p.inf.	2.22×10^{-4}
Geometric(1/2)	11/10	4.59	4.59	4.72×10^{-10}	p.inf.	2.26×10^{-3}
Geometric(−1/2)	1/2	p.inf.	2.29×10^{-5}	1.43×10^{-5}	0.420	1.53×10^{-5}
Geometric(−1/2)	1	1.00	1.00	2.67×10^{-4}	p.inf.	1.23×10^{-4}
Geometric(−1/2)	11/10	4.59	4.59	1.04×10^{-3}	p.inf.	1.37×10^{-3}
Triangle wave	1/2	1.00	1.00	956	1.72	1.00
Triangle wave	1	n/a	1.50	0.703	242	0.500
Triangle wave	11/10	4.87	4.77	0.677	p.inf.	0.524
Polynomial(1/t)	1/2	0.0625	0.0625	71.1	0.679	6.29×10^{-2}
Polynomial(1/t)	1	n/a	1.02	9.65×10^{-3}	p.inf.	8.20×10^{-4}
Polynomial(1/t)	11/10	4.60	4.60	1.65×10^{-3}	p.inf.	4.44×10^{-3}
Polynomial(t)	1/2	31.0	31.0	p.inf.	36.3	31.0
Polynomial(t)	1	n/a	34.5	16.9	p.inf.	1.08×10^{-4}
Polynomial(t)	11/10	83.2	76.8	4.05	p.inf.	0.367
Uniform(5,10)	1/2	1.53×10^{-5}	1.53×10^{-5}	4.39×10^{-4}	5.78×10^{-3}	2.73×10^{-5}
Uniform(5,10)	1	1.0	1.0	4.06×10^{-2}	0.437	2.65×10^{-3}
Uniform(5,10)	11/10	4.59	4.59	0.119	0.814	5.34×10^{-3}

Table 8 contains the errors with parameters $T_{\max} = 512$, $N = 256$ and $a = 1$. At this parameter setting, all other methods except CMG fail, while CMG gives a reasonably good approximation.

Table 8. Errors for various test functions ($N = 256$, $T_{\max} = 512$, $a = 1$).

Function	CIR	CIS	Ort1	Ort2	CMG
Dirac(10)	1.00	1.00	p.inf.	p.inf.	1.51×10^{-3}
Poisson(1)	0.368	0.368	p.inf.	p.inf.	1.61×10^{-3}
Heaviside step	n/a	1.50	p.inf.	p.inf.	1.01×10^{-3}
Geometric(1/2)	1.00	1.00	p.inf.	p.inf.	1.23×10^{-4}
Geometric(−1/2)	1.00	1.00	p.inf.	p.inf.	1.23×10^{-4}
Triangle wave	n/a	1.50	p.inf.	p.inf.	0.500
Polynomial($1/t$)	n/a	1.0	p.inf.	p.inf.	6.11×10^{-3}
Polynomial(t)	n/a	575	p.inf.	p.inf.	9.29×10^{-4}
Uniform(5,10)	1.0	1.0	p.inf.	p.inf.	3.28×10^{-3}

8. Conclusions

In this paper, we collected many of the NIZT methods from the literature and presented two new methods. One of them, the CIS method, is a variant of the contour integral-based CIR method and the other one, the CMG method, is inherited from numerical inverse Laplace transformation.

A wide numerical investigation on these NIZT methods indicated that different methods are accurate when the order of the method is higher than the required parameter ($N > T$) and when it is lower ($N \leq T$). In the first case, the CIR and the CIS methods are the most reliable (moderate error and tolerable precision loss) and they perform rather similarly. Their behavior differs only when one of the methods must evaluate the transform function near one of its poles. In the second case, when $N \leq T$, the CMG method outperforms all other methods in both accuracy and precision loss. The rest of the methods perform poorly, in general, except the Ort1 method, which gives more accurate results than the CIR and the CIS methods if appropriately high precision is used, but its numerical instability often results in larger precision loss than the applied high precision arithmetic.

While the optimal method to choose may depend on multiple factors (e.g., tolerated precision loss, computational time, tolerated error, etc.) as a rule of thumb for $N > T$ we recommend to use the CIR and CIS methods, and for $N \leq T$ the CMG method. The Mathematica implementation of the considered NIZT methods and some results of the numerical evaluation are available at http://webspn.hit.bme.hu/~telek/tools/nizt.zip.

Author Contributions: The contribution is equally shared by the authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially supported by the OTKA K-123914 and the TUDFO/51757/2019-ITM projects.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Oppenheim, A.V.; Schafer, R.W. *Discrete-Time Signal Processing*, 3rd ed.; Pearson: London, UK, 2009.
2. Abate, J.; Choudhury, G.L.; Whitt, W. An Introduction to Numerical Transform Inversion and Its Application to Probability Models. In *Computational Probability*; Grassmann, W.K., Ed.; Springer US: Boston, MA, USA, 2000; pp. 257–323. doi:10.1007/978-1-4757-4828-4_8. [CrossRef]
3. Mills, P.L. Numerical inversion of z-transforms with application to polymerization kinetics. *Comput. Chem.* **1987**, *11*, 137–151. [CrossRef]
4. Merrikh-Bayat, F. Two Methods for Numerical Inversion of the Z-Transform. *arXiv* **2014**, arXiv:1409.1727.
5. Tagliani, A. Inverse Z transform and moment problem. *Probab. Eng. Inf. Sci.* **2000**, *14*, 393–404. [CrossRef]
6. Rajković, P.M.; Stanković, M.S.; Marinković, S.D. A method for numerical evaluating of inverse Z-transform. *Facta Univ.-Ser. Mech. Autom. Control. Robot.* **2004**, *4*, 133–139.
7. Abate, J.; Whitt, W. Numerical inversion of probability generating functions. *Oper. Res. Lett.* **1992**, *12*, 245–251. doi:10.1016/0167-6377(92)90050-D. [CrossRef]

8. Cooley, J.W.; Tukey, J.W. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **1965**, *19*, 297–301. [CrossRef]
9. Rosetta Code. Fast Fourier Transform. 2019. Available online: https://rosettacode.org/wiki/Fast_Fourier_transform (accessed on 1 August 2019).
10. Duhamel, P.; Vetterli, M. Fast Fourier transforms: a tutorial review and a state of the art. *Signal Process.* **1990**, *19*, 259–299. [CrossRef]
11. Bluestein, L. A linear filtering approach to the computation of discrete Fourier transform. *IEEE Trans. Audio Electroacoust.* **1970**, *18*, 451–455. [CrossRef]
12. Horváth, G.; Horváth, I.; Almousa, S.A.D.; Telek, M. Numerical Inverse Laplace Transformation by concentrated matrix exponential distributions. *Perform. Eval.* **2020**, *137*, 102067. doi:10.1016/j.peva.2019.102067. [CrossRef]
13. Abate, J.; Whitt, W. A Unified Framework for Numerically Inverting Laplace Transforms. *INFORMS J. Comput.* **2006**, *18*, 408–421. [CrossRef]
14. Asmussen, S.; Bladt, M. Renewal Theory and Queueing Algorithms for Matrix-Exponential Distributions. In *Lecture Notes in Pure and Applied Mathematics*; Alfa, A.S., Chakravarthy, S.R., Eds.; Marcel Dekker: New York, NY, USA, 1997; pp. 313–341.
15. Horváth, G.; Horváth, I.; Telek, M. High order concentrated matrix-exponential distributions. *Stoch. Model.* **2019**. doi:10.1080/15326349.2019.1702058. [CrossRef]
16. inverselaplace.org. Available online: http://inverselaplace.org/ (accessed on 13 February 2020).