LONDON'S GLOBAL UNIVERSITY

Beyond the Circle: Deforming Contours in Inverse Z-Transform

Final Year Project Report

Roman Ryan Karim[1]

Dr Carolyn Phelan

Department of Computer Science
University College London

Submission date: March 23, 2024

[1] **Disclaimer:** This report is submitted as part requirement for the MEng degree in Mathematical Computation at UCL. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract

Contents

Todo list

Chapter 1

Introduction

1.1 Motivation

1.2 Aims and Objectives

1.3 Overview

Chapter 2

Background

In Chapter 2, we establish a foundational understanding of the topic in hand. This section is designed to be self-contained, providing essential background for all readers, while references are included for those seeking a deeper exploration.

"By definition, a complex number z is an ordered pair (x, y) of real numbers x and y, written $z = (x, y)$" (Kreyszig, 2010). In practice, complex numbers are written in the form $z = x + iy$, where x and y are real numbers and i is the imaginary unit. We may find it easier to represent complex numbers in their polar form, $z = re^{i\theta}$, where r represents the magnitude of z and θ represents the angle of z with respect to the positive real axis. The set of complex numbers is denoted by $\mathbb{C}$.

2.1 The $\mathscr{Z}$-Transform

The z-transform is a transformation of a real or complex time function $x(n)$, often used for analyzing discrete-time signals and systems. It is a generalization of the discrete-time Fourier transform (DTFT) that extends the analysis to the complex plane. The Z-transform is formally defined as:

$$X(z) = \mathscr{Z}_{n \to z}[x(n)] = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \tag{2.1}$$

For a convenient description of z in the complex plane, we tend to its polar form $z = re^{i\theta}$.

In the analysis of causal systems - systems for which a time origin is defined and is illogical to consider signal values for negative time - the unilateral z-transform is used. Unlike the bilateral z-transform in Eq. (2.1), we sum from zero to positive infinity yielding

$$X(z) = \mathscr{Z}_{n \to z}[x(n)] = \sum_{n=0}^{\infty} x(n) z^{-n} \tag{2.2}$$

The region within the complex z-plane where the z-transform converges is known as the Region of Convergence (ROC). The ROC is defined for the set of values of z for which the z-transform is absolutely summable

$$\mathbf{ROC} = \left\{ z : \sum_{n=0}^{\infty} |x(n) z^{-n}| < \infty \right\} \tag{2.3}$$

For causal sequences, the ROC is typically the exterior of the outermost pole in the Z-plane, denoted as $|z| > a$. If we say that z_1 converges, then z_1 exists within the ROC. Thus, all z such that $|z| \geq |z_1|$ also converge. This region excludes the poles themselves, as the transform does not converge at those points. For the system to be *stable*, the ROC must include the unit circle, $|z| = 1$, implying that all poles must lie within the unit circle (Loveless and Germano, 2021).

2.1.1 Relation to the Fourier Transform

It is useful to note the relationship between the z-transform and the Fourier transform. Taking the Fourier transform of a sampled function $x(t)$ results in:

$$\mathcal{F}\left[x(t)\sum_{n=-\infty}^{\infty}\delta(t-n\Delta t)\right] = \int_{-\infty}^{\infty} x(t)\sum_{n=-\infty}^{\infty}\delta(t-n\Delta t)e^{-i\omega t}dt \tag{2.4}$$

$$= \sum_{n=-\infty}^{\infty}\int_{-\infty}^{\infty} x(t)\delta(t-n\Delta t)e^{-i\omega t}dt \tag{2.5}$$

$$= \sum_{n=-\infty}^{\infty} x(n\Delta t)e^{-i\omega nt} \tag{2.6}$$

where we make use of the sifting property of the delta function. If we normalize the sampling interval to 1, we get

$$\sum_{n=-\infty}^{\infty} x(n)e^{-in\omega} \tag{2.7}$$

This is the discrete-time Fourier transform (DTFT) of the sequence $x(n)$. The sequence $x(n)$ is sampled at discrete-time intervals $t_n = n\triangle t$, where the sampling interval $\triangle t$ is the time between consecutive samples and the time index n numbers the samples. The DTFT is a periodic function of ω with period 2π, and its existence relies on the absolute summability of the sequence $x(n)$:

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \tag{2.8}$$

The Z-transform generalizes Eq. (2.7) to the complex plane, not just the unit circle where $r = 1$ (Schafer and Oppenheim, 1989).

maybe write a bit more on why we need to generalize it?

2.1.2 Relation to Probability Distribution Functions

Understanding the behavior of random events and signals is crucial in various fields, including finance, engineering, and computer science. This understanding is deepened through the study of Probability Distribution Functions; the Probability Mass Function (PMF) for discrete random variables and the Probability Density Function (PDF) for continuous random variables. Given the nature of this project, we'll be focusing our attention on the PMF.

The PMF is defined for a discrete random variable X taking on values x_i with probabilities p_i, as $P(X = x_i) = p_i$. The PMF satisfies the following properties:

$$\sum_{i=0}^{n} p_i = 1 \qquad \text{and} \qquad 0 \leq p_i \leq 1 \ \forall i \tag{2.9}$$

Example 1 *Consider a fair six-sided dice. The PMF for the dice roll is given by*

$$p(x) = \begin{cases} \frac{1}{6} & \text{if } x = 1, 2, 3, 4, 5, 6 \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

where $p(x)$ is the probability of rolling a number x.

more research needed to bridge the gap

We can then expand upon the PMF, $p(x)$, to obtain the Probability Generating Function (PGF), $G_X(q)$, defined as

$$G_X(q) = E[q^X] = \sum_{x=0}^{\infty} p(x)q^x, \tag{2.11}$$

where $E[\cdot]$ denotes the expectation operator, and q is a complex number. We deliberately use q to distinguish the PGF from the z-transform.

The concept of summarizing information is not unique to probability theory. In the analysis of signals, we aim to encapsulate the behaviour of a sequence into a single function. This is akin to the PGF, where the z-transform is used to analyze discrete-time signals and systems. Drawing on the principles outlined by Ross (2014), we can bridge the gap between probability theory and signal processing, leveraging the z-transform to analyze the behaviour of signals in the complex plane.

2.2 The Inverse $\mathcal{Z}$-Transform

The inverse Z-transform aims to find the n-th value of the sequence $x(n)$ given the Z-transform $X(z)$. This is commonly defined as a Cauchy integral around a contour C in the complex plane. The contour C is a counter-clockwise closed path that encloses the region of convergence (ROC). The inverse Z-transform is formally given by

$$x(n) = \mathcal{Z}^{-1}_{z \to n}[X(z)] = \frac{1}{2\pi i} \oint_C X(z) z^{n-1} dz \tag{2.12}$$

In real-world applications, we often require numerical approximation due to computational challenges posed by the Cauchy integral formula. Such approximations enable the effective analysis and processing of complex signals within various technological and financial systems.

2.2.1 Abate and Whitt 1992

The numerical approximation formula offered by Abate and Whitt (1992a,b) is based on a Fourier series catering to the inversion of probability generating functions as elucidated in Section 2.1.2. The format is conducive to queuing theory and other probabilistic models where the Z-transform is defined as $q = 1/z$. The authors approximate the inversion using a trapezoidal rule for numerical integration over a complex contour given by

$$x(n) \approx \frac{1}{2nr^n} \left(X(r) + 2 \sum_{k=1}^{n-1} (-1)^k \mathrm{Re}\left(X(re^{\frac{ik\pi}{n}}) \right) + (-1)^n X(-r) \right) \tag{2.13}$$

The parameter r is used to control the error; setting $r = 10^{-\lambda/2n}$ yields an accuracy bound of $10^{-\lambda}$. The authors leverage the inherent symmetry within the complex plane to enhance computational efficiency by exploiting the complex conjugate symmetry of $X(z)$; each term $X(re^{\frac{ik\pi}{n}})$ in the upper half has a mirror image in the lower half. The computational load is thus halved by *folding* the problem in this manner.

Given the nature of this project, we may find it easier to use the following definition, where we set $z = 1/q$, to approximate Eq. (2.12).

$$x(n) \approx \frac{1}{2nr^n}\left(X(\frac{1}{r}) + 2\sum_{k=1}^{n-1}(-1)^k \mathrm{Re}\left(X(\frac{1}{re^{\frac{ik\pi}{n}}})\right) + (-1)^n X(-\frac{1}{r})\right) \tag{2.14}$$

talk about the Nyquist-Shannon sampling theorem? - number of points must be double to avoid aliasing

2.2.2 Cavers 1978

Extending upon our analysis in Section 2.1.1, Cavers (1978) proposes to sample the z-transform of a function on a circular contour at equally spaced points and then apply the inverse FFT to these sampled points to approximate the original time-domain signal. We can formulate this as:

$$f(n) = \frac{r^n}{N}\mathrm{IFFT}[f(re^{2\pi i/N})] \tag{2.15}$$

2.2.3 Series acceleration techniques

2.3 Pricing Options

2.3.1 Discrete Monitoring

2.4 Optimization Techniques

In the context of computational mathematics, optimizations techniques are used to identify the optimal or a sufficiently effective solution to a problem within a given set of constraints. The goal is to minimize or maximize a specific objective function by systematically choosing the values of the variables. The objective function is often referred to as the *cost function* or *loss function* and the variables are referred to as *parameters*. The optimization problem can be formulated as

$$\text{minimize } f(x) \text{ subject to } x \in \Omega \tag{2.16}$$

where $f(x)$ is the objective function and Ω is the feasible region defined by the constraints of the problem.

Gradient descent is one of the most popular algorithms for parameter optimization with success in Deep Learning and Neural Networks employing variants of the algorithm (Lu and Jin, 2017; Zhang, 2019; Zeebaree et al., 2019). The adaptability to diverse problem domains (Tian et al., 2023) parallels our use case, where gradient descent is applied outside traditional deep learning to optimize parameters of a mathematical function (Persson et al., 2022). This reinforces the potential of gradient descent algorithms in broader computational mathematics, affirming its efficacy in finding optimal solutions in complex optimization landscapes.

2.4.1 Gradient Descent

Gradient descent iteratively converges to a local minimum of a function by moving in the direction of the steepest descent, as defined by the negative gradient. This method is expressed mathematically as

$$x_{k+1} = x_k - \alpha_k \nabla f(x_k) \tag{2.17}$$

where x_k is the parameter vector at iteration k, α_k is the learning rate, and $\nabla f(k)$ represents the gradient of the function at x_k. The selection of α_k is crucial as it determines the size of the step taken towards the minimum; too large can overshoot the minimum, too small can result in a long convergence time. The process repeats until a predetermined termination criterion is met, typically when the change in the value of $f(k)$ falls below a threshold. This iterative process is showcased in the pseudocode below:

Algorithm 1 Gradient Descent

```
Initialize x_0, set k = 0
while termination conditions not met do
    Compute gradient ∇f(x_k)
    Choose a suitable step size α_k
    Update x_{k+1} = x_k − α_k ∇f(x_k)
    k = k + 1
end while
```

Stochastic Gradient Descent

However, classic Gradient Descent faces limitations, including susceptibility to local minima and potential for overshooting or long convergence times. Stochastic Gradient Descent (SGD) addresses these issues by introducing variability in the optimization process. It modifies Eq. (2.17) to use a randomly selected subset of data to compute the gradient, to allow for dynamic adjustment of the learning rate and leveraging noise to escape local minima. We define the update rule to

$$x_{k+1} = x_k - \alpha_k \nabla f_{i_k}(x_k) \tag{2.18}$$

where $\nabla f_{i_k}(x_k)$ is the gradient of the cost function with respect to a random subset i_k. We thus avoid the pitfalls associated with a static learning rate and promotes a quicker convergence time.

include diagram to show faster convergence time (illustration purposes)

Chapter 3

Experiment

Finding different parameters to use for the experiment making use of Machine Learning techniques.

Chapter 4

Results

Chapter 5

Conclusion

5.1 Summary

5.2 Future Work

5.3 Acknowledgements

Bibliography

Abate, J. and Whitt, W. (1992a). The fourier-series method for inverting transforms of probability distributions. *Queueing systems*, 10:5–87.

Abate, J. and Whitt, W. (1992b). Numerical inversion of probability generating functions. *Operations Research Letters*, 12(4):245–251.

Cavers, J. (1978). On the fast fourier transform inversion of probability generating functions. *IMA Journal of Applied Mathematics*, 22(3):275–282.

Kreyszig, E. (2010). *Advanced Engineering Mathematics*. John Wiley & Sons.

Loveless, B. and Germano, G. (2021). Review of numerical inversion techniques of the z-transform. Working Paper.

Lu, S. and Jin, Z. (2017). Improved stochastic gradient descent algorithm for svm. *International Journal of Recent Engineering Science (IJRES)*, 4(4):28–31.

Persson, P.-O., Franco, M., and Sweeney Blanco, R. (2022). Gradient-based optimization. Accessed: [10-03-2024].

Ross, S. M. (2014). *Introduction to probability models*. Academic press.

Schafer, R. W. and Oppenheim, A. V. (1989). *Discrete-time signal processing*, volume 5. Prentice Hall Englewood Cliffs, NJ.

Tian, Y., Zhang, Y., and Zhang, H. (2023). Recent advances in stochastic gradient descent in deep learning. *Mathematics*, 11(3).

Zeebaree, D. Q., Haron, H., Abdulazeez, A. M., and Zebari, D. A. (2019). Trainable model based on new uniform lbp feature to identify the risk of the breast cancer. In *2019 international conference on advanced science and engineering (ICOASE)*, pages 106–111. IEEE.

Zhang, J. (2019). Gradient descent based optimization algorithms for deep learning models training. *arXiv preprint arXiv:1903.03614*.

Appendices

Appendix A

Initial Project Plan

LONDON'S GLOBAL UNIVERSITY

Numerical Benchmarking on Inverse Z-Transform and Its Uses in Discrete Pricing Options

Project Plan

Roman Ryan Karim

Supervisor: Dr Carolyn Phelan

Department of Computer Science
University College London

Submission date: 16 November 2023

Chapter 1

Aims and Objectives

1.1 Aims

We aim to understand a new efficient method for numerical evaluation of the inverse Z-transform, which states to be faster and more accurate than the standard trapezoid rule. A specific area of applying this method would be to the pricing of discretely monitored exotic options, such as lookback and barrier options, and see how it compares to other methods; Abate and Whitt's approach, C. Cavers' method with Euler, Shanks and epsilon accelerations, etc.

1.2 Objectives

- Understanding Levendorskii's inverse Z-transform and the common numerical evaluation methods
- Implementing the function as a code
- Numerical benchmarking; average error, maximum error and CPU time
- Exploring its uses in discrete pricing options

1.3 Deliverables

- numerical benchmarking results to add to '*Review of numerical inversion techniques of the z-transform'* by Loveless and Germano
- results and implementation in regards to discrete pricing options (*Accurate numerical inverse z-transform and it's use in the Fourier-z pricing of discretely monitored path-dependent options* by Loveless, Phelan and Germano)

Chapter 2

Work Plan

2.1 Project Start $\rightarrow$ 30$^{\text{th}}$ November '23

- background reading on complex numbers & contour integration based methods, fourier transform, z-transform and its inverse, numerical approaches to inverse z-transform and pricing options (barrier and lookback options)
- coding implementation of Levendorskii's inverse z-transform

2.2 1$^{\text{st}}$ December '23 $\rightarrow$ 24$^{\text{th}}$ January '24

- preliminary research on Loveless' and Germano's *'Review of numerical inversion techniques of the z-transform'*
- understanding the other methods; AW, C, CEuler, CShanks and CEpsilon
- going over the different functions; Heaviside Step, Polynomial, Decaying Exp, Sinusodial
- reviewing the code for numerical benchmarking
- implementing it for Levendorskii's method
- begin work on interim report

2.3 24$^{\text{th}}$ January '24 $\rightarrow$ 15$^{\text{th}}$ March '24

- preliminary recap on discrete pricing options (barrier and lookback options) and the need for z-transform
- use-case in discrete pricing options
- start work on project report; however, to be worked on throughout the year/stages

2.4 5$^{\text{th}}$ March '24 $\rightarrow$ 26$^{\text{th}}$ April '24

- extra time to deal with any unexpected problems or delays
- final touches