MASTER

MATHEMATICAL FINANCE

MASTER'S FINAL WORK

REPORT

APPLIED MATHEMATICS IN DECENTRALIZED FINANCE

Ruben dos Reis Mendes

December 18, 2023

MASTER

MATHEMATICAL FINANCE

MASTER'S FINAL WORK

REPORT

APPLIED MATHEMATICS IN DECENTRALIZED FINANCE

Comparative Analysis of Impermanent Loss Dynamics: Unraveling the Mechanisms in Balancer, Uniswap, and Curve Finance

Ruben dos Reis Mendes

Supervisors:

Prof. João Paulo Janela

Dr. Carolina Goldstein

December 18, 2023

Acknowledgments

I would like to express my sincere gratitude to the following individuals and organizations for their support, guidance, and contributions throughout the completion of this work:

- Prof. João Miguel Guerra: For his invaluable guidance and mentorship throughout the research process.
- Dr. Tiago Martins and Dr. João Farinha: For their valuable discussions, insights, and assistance.
- Prof. João Paulo Janela and Dr. Carolina Goldstein: For their constructive feedback and support.
- Exclusive Dialogue, Lda: For providing the necessary resources and environment for this research.
- My friends and family: For their encouragement and understanding during this academic journey.

I extend my heartfelt appreciation to all of you for your contributions and support in making this work a reality.

Abstract

Within the realm of decentralized finance, liquidity providers are esteemed for their vital role in maintaining liquidity pools. Nevertheless, they are exposed to a significant risk referred to as impermanent loss. This happens when the prices of one or more tokens fluctuate in relation to others or others within the same liquidity pool, ultimately reducing the initial quantity of one or more assets and leading to a temporary loss for the liquidity provider.

Impermanent loss can be affected by several factors, including price volatility, asset correlation, trading volume, fees earned, time, and pool size. To reduce the impact of impermanent loss in different market conditions, it is important to have a good understanding of these factors and to choose the appropriate liquidity provision strategies. By doing so, one can minimize the negative effects of impermanent loss.

To make informed investment decisions, liquidity providers must consider impermanent loss and carefully choose the most advantageous DeFi protocol to provide liquidity. This enables effective investment management and helps determine whether to enter or exit a specific pool. This research provides a risk assessment that offers liquidity providers guidelines to evaluate which of the four DeFi protocols is likely to be the most optimal choice in terms of impermanent loss.

We start by providing an overview of how the Bitcoin blockchain works and then focus our discussion on the Balancer protocol, Uniswap, and Curve Finance. Throughout the research, we also stressed the significance of these decentralized exchange's Value functions and price definitions for the corresponding impermanent loss formula.

From the premisses outlined in the Balancer, Uniswap V2, V3, and Curve Finance whitepapers, we establish and prove the correspondent impermanent loss formula to be used throughout our discussion leaving the Curve Finance section in the Appendix due to the page number restriction suggested by the School. Considering that to calculate the impermanent loss, one needs to have pool token prices at a maturity date, we used a pure jump Lévy stochastic process to model the token log price dynamics, which allows us to estimate any token price at a maturity date. Since any market model based on a Lévy process is complex by nature, we also provide a careful study of each stochastic process involved in the construction of our model.

At last, we apply the derived model to a specific case, and once instances of our model are calibrated to the time series of each token, we manipulate these parameters to generate distinct market conditions, notably those that are most prevalent. Subsequently, we conducted a comparative analysis of the corresponding impermanent losses within these protocols. In this manner, it became feasible to estimate in which of the four decentralized finance protocols the liquidity provider's investment would be better shielded against impermanent loss.

Keywords: Bitcoin Blockchain, Ethereum Blockchain, Decentralized Finance, Decentralized Exchange, Value Function, Impermanent Loss.

Resumo

Os provedores de liquidez, ao contribuir com ativos para pools de liquidez, desempenham um papel importante na finanças descentralizada. No entanto, estão vulneráveis a um risco significativo conhecido como perda não realizada (PNR). Isso acontece quando os preços de um ou mais tokens flutuam em relação aos outros ou a outros dentro do mesmo pool de liquidez, reduzindo, em última instância, a quantidade inicial de um ou mais ativos e resultando em uma PNR para o provedor de liquidez.

A PNR pode ser afetada por diversos fatores, incluindo volatilidade de preços, correlação de ativos, volume de negociação, taxas auferidas, tempo e tamanho da pool. Para mitigar o impacto da perda não realizada em diferentes condições de mercado, é crucial possuir uma compreensão aprofundada desses fatores e selecionar estratégias adequadas de provisão de liquidez. Dessa forma, é possível minimizar os efeitos negativos da perda não realizada.

Para tomar decisões de investimento informadas, os provedores de liquidez devem considerar a perda não realizada e escolher cuidadosamente o protocolo de Finaças Descentralizada (FiD) mais vantajoso para fornecer liquidez. Isso possibilita uma gestão eficaz dos investimentos e auxilia na decisão de entrar ou sair de uma pool específica. Este estudo fornece uma análise de risco que oferece aos provedores de liquidez diretrizes para avaliar qual dos quatro protocolos FiD é provavelmente a escolha mais otimizada em termos da PNR.

Iniciamos nossa exploração fornecendo uma exposição sobre os mecanismos operacionais da blockchain do Bitcoin, direcionando posteriormente nosso foco para a análise do protocolo Balancer, Uniswap e Curve Finance. Ao longo de nossa discussão, enfatizamos a importância da formulações das Funções de Valor, assim como as definições das taxas marginais de substituição inerentes a cada um dessas corretoras descentralizadas, na dedução das fórmulas que nos facultam a PNR como função de tempo.

Com base nas premissas delineadas nos whitepapers do Balancer, Uniswap V2, V3 e Curve Finance, estabelecemos e comprovamos a fórmula correspondente de perda não realizada a ser utilizada ao longo da nossa discussão deixando a seção do protocolo Curve

Finance no Apêndice devido à restrição do úmero de página sugeridas pela Escola. Tendo em consideração que para calcular a perda não realizada é necessário ter os preços dos ‘tokens’ da pool na maturidade, utilizamos um modelo de mercado de Lévy baseado no processo estocástico Variância Gama processo que corresponde a um processo de salto puro com o intuito de modelar a dinâmica do logaritmo de um ‘token’ qualquer.

Repare que qualquer modelo de mercado baseado num processo de Lévy é complexo por natureza. Com efeito, também apresentamos um estudo detalhado de cada processo estocástico envolvido na construção do nosso modelo.

Por fim, aplicamos o modelo derivado a um caso específico e, uma vez calibrado instâncias do nosso modelo às séries temporais de cada um dos ‘token’, manipulamos estes parâmetros para originar diferentes condições do mercado, nomeadamente as mais usuais. Seguidamente, fizemos uma análise comparativa das correspondentes perdas não realizadas nestes protocolos. Desta forma foi possível estimar em qual dos quatro protocolos da finança descentralizada o investimento deste provedor de liquidez estaria melhor protegido relativamente à PNR.

Palavras-chave: “Blockchain” de “Bitcoin”, “Blockchain” de Ethereum, Finanças Descentralizada, Corretoras Descentralizada, Função de Valor, Perda Não Realizada.

Contents

1 Introduction

Contextualization

The Blockchain technology has captured significant attention recently due to its revolutionary nature. Essentially, a blockchain consists of a sequence of blocks, each containing transactions, organized in a predetermined order. The blocks serve as the essential components, with the first block acting as the foundation, and to ensure the security of the blockchain, each block receives a unique cryptographic hash. Moreover, each block includes a reference to the previous block, creating a sequence of hashes that can be traced back to the original block, known as the genesis block. The ”previous block hash” section in the block header is important to the blockchain's integrity. Any modification to a parent block affects its hash, which in turn affects the hash of the child block. This cascading effect extends to following blocks, making it computationally difficult to change earlier blocks without recalculating the whole chain. As a result, the blockchain's extensive history becomes immutable, providing a key security component in systems like Bitcoin.

A Decentralized Application (DApp) is an innovative type of application that operates on a decentralized blockchain network. Unlike traditional applications, DApps are built to function in a decentralized environment, meaning that they are not controlled by any single entity or authority. DApps are created through blockchain technology, which offers enhanced security, transparency, and immutability.

Popular decentralized blockchain networks like Ethereum give developers the necessary tools and infrastructure to build DApps. These DApps are designed to be open-source and trustless, meaning that users can access or interact with the application without relying on intermediaries. Transactions are processed on the blockchain network, and all participants can view these transactions in real-time, ensuring transparency. DApps can be utilized for a wide range of purposes, including financial applications, social networks, gaming, and much more (see [3], and [15]).

An Automated Market Maker (AMM) falls under the category of decentralized exchanges (DEX), specifically a type of DApp that uses a mathematical algorithm to determine the price of assets. Unlike centralized exchanges that rely on order books and matching engines, AMMs execute trades using smart contracts in a decentralized and automated fashion. To function, AMMs require liquidity pools that are established by users whose tokens deposit into a smart contract. These tokens allow for trades between different assets, with the AMM adjusting the token price based on asset supply and demand (see [11]).

Decentralized Finance (DeFi) is a new financial system that allows individuals to access various financial services without the need for traditional financial intermediaries such as banks, brokerages, or other financial institutions. DeFi utilizes blockchain technology to create a peer-to-peer financial system that is transparent, open, and accessible to anyone with an internet connection.

DeFi is an innovative approach to finance because it eliminates the need for intermediaries and enables users to have complete control over their assets. This decentralized nature also allows DeFi to operate 24/7, without any restrictions on location. DeFi protocols are built on blockchain technology, which ensures reliability and security.

Two examples of DeFi protocols are Balancer and Curve Finance. As a decentralized exchange, Balancer uses a unique AMM system that ensures liquidity and price efficiency for all users. Balancer also allows users to create their own customized pools of tokens, which can be used for trading, staking, or lending (see [10]).

Curve Finance, on the other hand, is a decentralized exchange that is specifically designed for stablecoins. It allows users to trade various stablecoins, such as USDT, DAI, and USDC, with minimal slippage and low fees. Curve Finance also offers users the ability to earn rewards by providing liquidity to the platform (see [7]).

We will briefly discuss the Bitcoin blockchain on a conceptual level, explaining how it works by analyzing the transaction (TX) life cycle, e.g., the process that takes place from the creation of a TX up until the ledger in which it is included is confirmed into the chain (see [4]). Afterwards, we will make a brief introduction to the Ethereum Blockchain and highlight its innovative particulars, then discuss some specialized applications built upon it. The assimilation of the fundamental concepts presented in Appendix A and processes will be essential to fully understand the risk assessment, which we will elaborate on in our discussion.

Scope of the research

Liquidity Providers (LPs) that lend tokens to a Decentralized Exchange (DEx) face a variety of risks, including Impermanent Loss (IL) attributed to a volatile market. LPs benefit from data-driven insights provided by mathematical models, which allow them to make informed decisions, improve strategies, and limit risks.

Cryptocurrency prices can be very volatile, exposing LPs to Impermanent losses as token values fluctuate within the liquidity pool. This research aims to create a reliable mathematical model that assists LPs in navigating the complexities of DeFi liquidity provision. By analyzing IL risk, this model provides insights into optimized strategies.

To commence, we will develop the IL formula as a function of initial token prices and their respective prices at the maturity time horizon and each token's weight in the pool (in the case of the Balancer protocol). Once derived, we will provide a brief introduction to the particular type of Lévy stochastic process formally known as the Variance Gamma process as a means to obtain any number of future trajectories of a given token price. This section will be of utmost importance once it allows us to apply the Monte Carlo method to estimate any token price up until a considered maturity horizon.

This project is a result of the internship developed at a blockchain engineering and research consulting firm Exclusive Dialogue, usually referred to as ThreeSigma. In May of 2022, three outstanding engineers, Afonso Oliveira, Eduardo Morgado, and Tiago Barbosa, from one of Europe's finest engineering schools, Instituto Superior Técnico, founded the company. The company operates into three departments: Blockchain engineering, where they provide end-to-end services such as blockchain implementation, tailored solution setup, architectural design, and continuous maintenance. They have worked on projects such as Starkware, Arc 77-Bit, and many more.

Yeti Finance and the Avantis protocol are only two of the numerous initiatives on which the Economic Modeling department has worked extensively. Finally, the Code Audits department offers an efficient smart contract security auditing service to ensure your application is secure and ready for launch.

2 Theoretical Foundations

The Bitcoin infrastructure is based on a Peer-to-peer (P2P) network, meaning that all the computers (nodes) that participate in the network have equal status. There are no superior nodes, and every computer shares the responsibility of providing network services. The nodes in the network connect in a mesh topology, which means there are no hierarchical structures. In a P2P network, there is no central server or hierarchy. Participants both provide and consume services with reciprocity as the incentive. This makes P2P networks decentralized, open, and resilient.

The Bitcoin network relies heavily on the "mempool", which stands for "memory pool." It serves as a repository for unconfirmed or pending transactions. When a Bitcoin transaction is conducted, it is first routed via the mempool before being included in a blockchain block.

Unspent Transaction Output (UTXO) is a crucial concept in blockchain-based cryptocurrencies, particularly Bitcoin. It represents the unspent portions of Bitcoin transactions and is a fundamental requirement for tracking account balances. UTXO play a significant role in enhancing blockchain security, privacy, and consensus by allowing users to verify transactions while keeping their total balance private (see [4]).

2.1 Node Types and Functionalities

The only way in which an individual can interact with a decentralized blockchain network is through the use of a node. A node in the Bitcoin network is a computer or any hardware that runs the Bitcoin software. These nodes communicate with each other to transmit and receive transactions, as well as to verify their authenticity based on the consensus rules and mechanisms in place. Though all nodes in the Bitcoin P2P network are equal, their roles may differ depending on the function they serve. A Bitcoin node is a combination of various functions, including routing, the blockchain database, mining, and wallet services. For example:

i) **Routing Function**: each Bitcoin node has a routing functionality that enables it to interact with other nodes and operate within the network. This function allows the node to establish connections with its peers and maintain them, which in turn facilitates the delivery of transactions and blocks across the network.

ii) **Validation and Propagation**: Bitcoin nodes play a crucial role in validating transactions and blocks to ensure their integrity and compliance with the network's policies. Prior to transmitting transactions to other nodes in the network, they verify their legality and ensure that they meet the necessary standards or consensus rules. This process is vital in maintaining the security and reliability of the Bitcoin network.

iii) **Full Nodes**: full nodes maintain an up-to-date copy of the Bitcoin blockchain. Full nodes can independently and authoritatively validate each transaction without relying on external sources. They receive network notifications of new blocks, which they validate and attach to their local blockchain copy.

Market participants and entities use full nodes in the Bitcoin network for various purposes. Identifying the key players who typically employ these nodes sheds light on the network's complex interactions. For instance:

iii.a) **Miners**: Bitcoin miners frequently utilize full nodes, and the operation of a full node empowers miners to autonomously ascertain the authenticity of transactions before their inclusion in the block they are endeavoring to mine. By employing this approach, miners can ensure the integrity of the blockchain network and thereby enhance the security of the entire system.

iii.b) **Exchanges**: Cryptocurrency exchanges that engage in Bitcoin transactions may opt to operate full nodes as a means of independently verifying and confirming incoming transactions.

iii.c) **Wallet Providers**: Bitcoin wallet providers and related services may choose to operate full nodes in order to augment the security and dependability of their wallets.

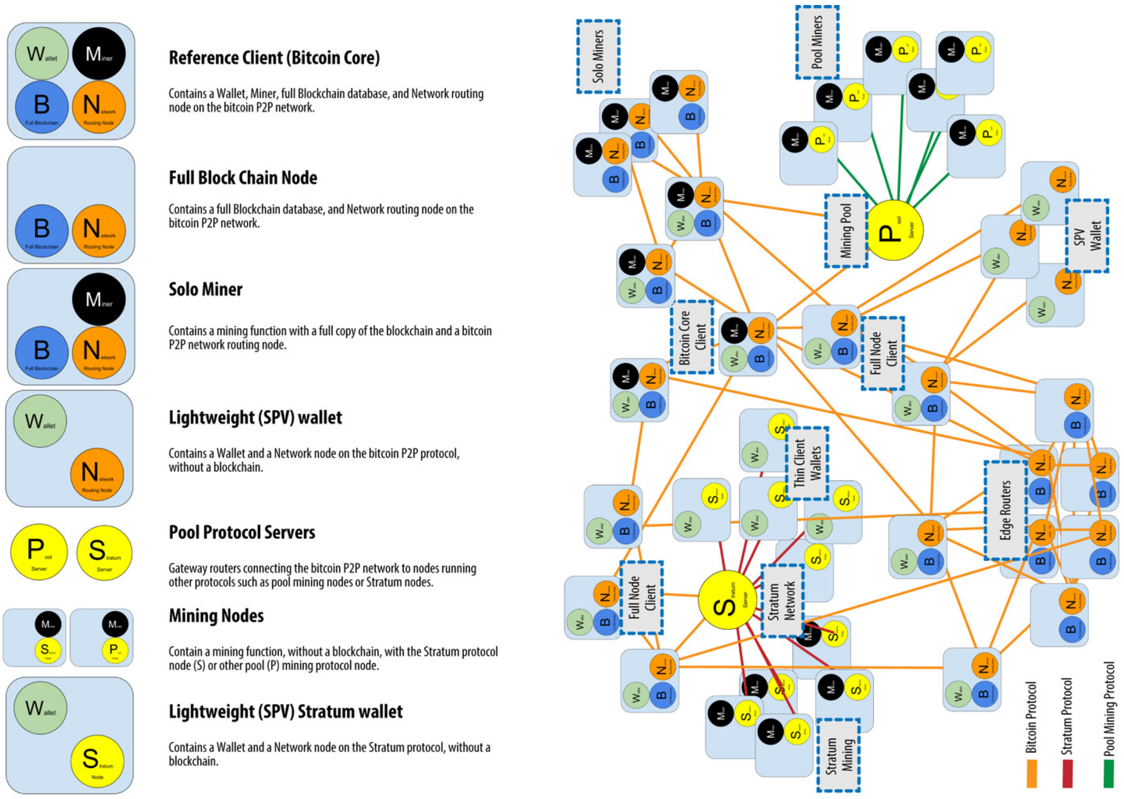
iv) The **Simple Payment Verification** (SPV) Nodes: also known as lightweight nodes, save only a portion of the blockchain rather than the entire copy. To confirm transactions, they employ a method known as rapid payment verification.
SPV nodes are given a filtered list of transactions associated with the addresses stored in their wallet. Here are some typical users of SPV Bitcoin nodes:

iv.a) **Mobile Wallets**: Mobile wallet applications frequently implement Simplified Payment Verification (SPV) nodes to offer users a more lightweight experience. This verification technique enables wallets to authenticate transactions without requiring the download of the complete blockchain, rendering them more appropriate for mobile devices with restricted storage and processing capabilities.

iv.b) **Point-of-Sale Systems**: It has been observed that businesses operating at physical locations, such as retail stores or restaurants, have started accepting Bitcoin payments and are increasingly relying on SPV nodes for their point-of-sale systems. The utilization of SPV nodes has enabled the swift verification of transactions, thereby enabling merchants to accept payments without having to wait for confirmations from the entire network.

v) **Mining** Nodes: mining nodes help create new blocks by solving the Proof-of-Work algorithm. Complete nodes are mining nodes that seek to mine new blocks and maintain an exact copy of the blockchain.

A Bitcoin node is a crucial component of the Bitcoin network, performing tasks such as routing, transaction validation, block validation, and blockchain maintenance. Apart from cryptocurrencies, blockchain technology has immense potential for applications in various sectors. Its decentralized and transparent nature allows for revolutionary changes in sectors such as banking, supply chain management, healthcare, and others. The following figure illustrates the different node types, protocols, and the extended Bitcoin network.



Reference Client (Bitcoin Core)

Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.

Full Block Chain Node

Contains a full Blockchain database, and Network routing node on the bitcoin P2P network.

Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.

Lightweight (SPV) wallet

Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

Pool Protocol Servers

Gateway routers connecting the bitcoin P2P network to nodes running other protocols such as pool mining nodes or Stratum nodes.

Mining Nodes

Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.

Lightweight (SPV) Stratum wallet

Contains a Wallet and a Network node on the Stratum protocol, without a blockchain.

(a) Node types.

(b) The Bitcoin Network sample.

2.2 Transaction Life Cycle in Bitcoin Network

The transaction, commonly referred to as TX, is the fundamental building block of the Bitcoin network. All other components within the network work together to ensure that transactions can be created, transmitted, verified, and ultimately added to the Bitcoin blockchain. In this section, we will provide a concise and easy-to-understand explanation of how this peer-to-peer network operates by tracing a transaction from its creation to the point where it becomes a part of the blockchain.

Mrs Gaimari, a businesswoman and CEO of a renowned tech enterprise, decided to surprise her CFO, Mrs Danica, with a bonus of 2,357 BTC after completing a significant project in the company. Mrs Gaimari used her smartphone to access her digital wallet application and then sent 2,357 BTC to Mrs Danica, ensuring that the transaction fee was generous enough to guarantee that her transaction would be processed quickly. Approximately ten minutes later, Mrs Danica received the 2,357 BTC in her wallet, and soon after, she received a message from Mrs Gaimari congratulating her on a well-deserved bonus.

But how 2,357 BTC was transferred from Mrs Gaimari´s wallet to Mrs Denica's wallet? The answer is provided through the following procedure:

Step 1

When Mrs Gaimari sent 2,357 BTC to her CFO firstly, Mrs Gaimari´s wallet created the TX by collecting enough UTXO of that particular key it controls adding the appropriate scripts and building new outputs (to the new owner and the change to be returned), sign that TX and once Mrs Gaimari pressed ≪send≫, her digital wallet broadcast that TX to all nodes connected to it through a process known as flooding.

Step 2

As they receive the TX, each node connected to Mrs Gaimari´s wallet checks if the TX structure is according to consensus rules at that moment, e.g.;

i) The transaction's syntax and data structure must be correct;

ii) The transaction size in bytes is less than MAX_BLOCK_SIZE;

iii) Each output value, as well as the total, must be within the allowed range of values (less than 21 million coins, more than the dust threshold);

iv) For each input, the referenced output must exist and cannot already be spent;

v) Neither lists of inputs or outputs are empty

vi) For each input, if the referenced output exists in any other transaction in the pool, the transaction must be rejected;

vii) A matching transaction in the pool, or in a block in the main branch, must exist...etc

After completing all the necessary verifications, this node adds the transaction to its mempool and then spreads it to all other connected nodes in the Bitcoin network. This process is repeated until every node in the network has Mrs Gaimari's transaction in their local mempool.

Step 3

When a miner node creates a candidate block, it selects the transaction (TX) with the highest fee from its local mempool and includes it in the block. This process is repeated until the block reaches its maximum size. It's important to note that the miner node also stays alert for blocks mined by other nodes in the network while building the candidate block. Once a valid block is detected, the competition to build the next block begins.

In order to receive compensation for winning the cryptographic battle to find the next block to be included in the blockchain, every miner node includes the coinbase transaction as the first transaction in their candidate block. Once a miner node successfully mines the block by finding a solution to the Proof-of-Work algorithm, it shares its candidate block with all other connected miner nodes.

Step 4

Nodes in a blockchain network verify each new solved block before sharing it with other nodes. This process ensures that only valid blocks are propagated across the network. It also guarantees that honest miners' blocks are added to the blockchain and rewarded accordingly. Dishonest miners, on the other hand, will have their blocks rejected and will lose not only the reward but also the computational work and electricity costs. When a node receives a new block, it validates it to ensure it complies with the consensus rule., e.g.:

i) The block data structure is syntactically valid;

ii) The block header hash is less than the target (enforces the Proof-of-Work)

iii) The block timestamp is less than two hours in the future (allowing for time errors);

iv) The block size is within acceptable limits

v) The first transaction (and only the first) is a coinbase Tx;

vi) All transactions within the block are valid.

All these criteria must be met for the block to be legitimate; otherwise, the block is rejected.

Step 5

Whenever a new block is received, a node tries to add it to the existing blockchain or parent chain. If the parent block belongs to the main chain, the new block extends the main chain. However, there are times when the new block extends a secondary chain. In such cases, the node compares the cumulative work of both chains. If the secondary chain has more work associated with it, the node switches to it as the new main chain. Eventually, all nodes achieve consensus by selecting the valid chain with the greatest cumulative work associated. Mining nodes vote with their mining power to extend a particular chain, resolving temporary discrepancies between competing chains.

Assuming that the recently added block to the Bitcoin blockchain contains Mrs Gaimari's transaction, the last step involves updating the UTXO set. Once this task is completed, Mrs Danica's wallet will detect that one of the keys it controls can now transact an additional 2,357 BTC.

To summarize, we have presented a description of the life cycle of a Bitcoin transaction that is sufficient for gaining a conceptual understanding of how the Bitcoin network operates. This process occurs rapidly, with a new block being added to the Bitcoin blockchain every ten minutes on average, containing between one thousand to seven thousand transactions. Transactions undergo a sequence of steps - initiation, confirmation, and permanent recording on the blockchain - while consensus rules and mechanisms ensure transparency and trust without any reliance on a central authority. Bitcoin's innovative architecture has opened up new financial possibilities while challenging established concepts of money and value exchange.

3 Ethereum Peer-to-Peer Network

Ethereum's peer-to-peer network was predominantly designed by Vitalik Buterin, a young programmer and Bitcoin enthusiast who wanted to expand Bitcoin's functionality and explore new possibilities beyond monetary applications, and Dr. Gavin Wood, a brilliant C++ programmer who went on to become a co-founder, co-designer, and CTO of Ethereum.

The main aim was to create a blockchain that could be used for a variety of purposes as a result of its programmability. By abstracting the nuances of peer-to-peer networks, consensus processes, and other underlying frameworks, Ethereum sought to develop a predictable and secure development environment for decentralised blockchain applications. In December 2013, Vitalik published a document titled Yellow Paper (see [15]) outlining the Ethereum concept: a quasi-Turing-complete (because of the gas constraint, the range of affordable computations is limited), and general-purpose blockchain. On July 30, 2015, the first Ethereum block was mined, officially launching the network.

Ethereum's peer-to-peer network serves as an open-source, globally decentralized computer architecture for executing smart contracts. This network employs blockchain technology to synchronize and preserve system state changes, and it leverages Ether, the native token, to assess and limit execution resource costs (see [3]).

While Ethereum and Bitcoin share features with prior open blockchains, such as a peer-to-peer network and a Byzantine fault-tolerant consensus mechanism (proof-of-stake for Ethereum), they also differ significantly. The fundamental purpose of Ethereum is not to create a digital currency payment network like Bitcoin. Ethereum is a programmable general-purpose blockchain that runs the Ethereum Virtual Machine (EVM) capable of running smart contracts of arbitrary and indefinite complexity. This means that, unlike Bitcoin's limited scripting language, Ethereum can function as a global computer with vast computing power.

3.1 *Ethereum Components*

1. **P2P network**

2. **Consensus rules** (defined in the yellow paper)

3. **Consensus mechanism**: Ethereum uses a consensus mechanism known as Gasper that combines Casper FFG proof-of-stake with the GHOST fork-choice rule.

4. **Economic security**

5. Transactions in Ethereum are composed by:

   Nonce - A sequence number, issued by the originating EOA, used to prevent message replay;

Gas price - The price of gas (in wei) the sender is willing to pay;

Gas limit - The maximum amount of gas the sender is willing to buy for this transaction;

Recipient - The destination Ethereum address;

Value - The amount of Ether amount to be send to the Recipient;

Data - The variable-length binary data payload;

v,r,s - The three components of an Elliptic Curve Digital Signature Algorithm (ECDSA) digital signature of the originating Externally Owned Account (EOA);

6. **State machine**: Ethereum state transitions are handled by the Ethereum Virtual computer (EVM), a stack-based virtual computer that executes bytecode (machine-language instructions). EVM programs, known as "smart contracts," are authored in high-level languages (such as Solidity) and compiled to bytecode for execution on the EVM;

7. **Data structures**: Ethereum's state is kept locally on each node in the form of a database (typically Google's LevelDB), which stores transactions and system state in a serialized hashed data structure known as a Merkle Patricia Tree;

8. **Clients**: Ethereum clients, which are software implementations of the Ethereum protocol, allow users to connect to the Ethereum blockchain. They greatly contribute to the network's decentralized character by providing several options for hosting a node. Example: Go-Ethereum (Geth) and Parity are two examples.

It is important to emphasize that in Ethereum network there are three main types of TX

1. The usual TX: a transaction from one EOA to another;

2. Contract deployment TX: a transaction without a Recipient, where the data field is used for the contract code;

3. Execution of a contract: a transaction that interacts with a deployed smart contract. In this case, Recipient address is the smart contract address.

Transaction life cycle in Ethereum

Earlier, we traced the life cycle of a Bitcoin network TX from its creation to the moment when the block carrying it was verified and included in the public ledger. The life cycle of an Ethereum network TX is similar, but there are some significant differences worth mentioning.

1. Smart Contracts: Ethereum network facilitates TXs that incorporate the implementation of smart contracts. These contracts are characterized as self-executing programs that are equipped with predetermined rules. This unique feature of Ethereum allows for more sophisticated and programmable transactions to take place, thereby expanding the breadth of possibilities within the network's ecosystem;

2. Ethereum Virtual Machine (EVM): Ethereum's system processes TXs using the EVM, which is responsible for executing smart contracts. This EVM boasts Turing-complete technology, making it capable of performing a broader range of computations than Bitcoin's scripting language;

3. Consensus Mechanism: the current consensus mechanism used by Ethereum is Proof of Stake (PoS). It is more energy-efficient, and enables faster TX processing times than the Bitcoin Proof of Work (PoW) consensus mechanism. Contrasting the Miners in the Bitcoin network in Ethereum network there are Validators which are chosen based on the amount of Ethereum they hold and are willing to "stake" as collateral. They create and validate new blocks and earn rewards in Ether (see [12]);

4. Gas Fees: tn the Ethereum network, a unit of measurement called "gas" is used to calculate the computational work and TX expenses. Users are required to pay gas fees in order to process TXs and execute smart contracts;

5. Confirmation Time: the confirmation times for Ethereum TX are generally faster when compared to Bitcoin. However, it is advisable for users to wait for multiple confirmations, especially for higher-value TXs or when interacting with DApps;

6. Transaction Reversibility: in some cases, Ethereum network TX can be more easily undone, particularly when dealing with smart contracts. This stands in stark contrast to the unalterable quality of Bitcoin transactions.

In essence, while Bitcoin and Ethereum both function as blockchain networks with similar TX life cycles, variations in their scripting capabilities, consensus mechanisms, and the role of smart contracts lead to differences in their TX life cycles. Ethereum's emphasis on smart contracts and programmability distinguishes it as a more flexible platform for DApps.

3.2 *Decentralized Application*

In our settings, a decentralized application (DApp) is a platform built on the Ethereum blockchain that offers multiple functionalities and services without depending on a central authority. This makes them more transparent, secure, and resistant to censorship. A DApp comprises at least one smart contract on a blockchain and a web-based user interface.

Examples of DApps on Ethereum:

i) **Aave** is a decentralized lending platform that allows users to borrow assets without the need for collateral, provided the loan is returned in full within the same transaction.

ii) **dYdX** is a decentralized trading platform with perpetual swaps that allows users to trade crypto derivatives with significant leverage.

iii) **Synthetix** is a DeFi protocol that allows users to create synthetic assets that mirror real-world assets like stocks, commodities, and fiat currencies. These products can be traded on the platform without the need to own the underlying assets.

The aforementioned instances highlight the scope and complexity of the Ethereum blockchain's DeFi domain. DeFi applications continue to transform the financial landscape by offering consumers alternative financial tools, income opportunities, and better control over their financial assets.

3.3 *Decentralized Exchange*

Ethereum offers a unique type of exchange called a decentralised exchange (DEX). Unlike traditional exchanges, DEXs are completely decentralised and trustless. They allow for direct, peer-to-peer transactions of digital assets without any middlemen involved. The Constant Function Market Maker (CFMM) method is a popular technology used in DEXs. It has brought significant improvements over the previous order book technology and represents a huge leap in the evolution of DEXs.

Let's consider a scenario where a trader wants to exchange Token A for Token B on a DEX. In a conventional order book exchange, the trader has to place a limit order indicating the quantity of Token B they need for the amount of Token A they are willing to trade. The transaction will only be executed when another user places an order that matches the trader's order.

However, the CFMM concept allows the DEX to operate based on a liquidity pool that includes reserves of both tokens. The asset-to-asset ratio of the pool remains constant. When a trader wants to swap tokens, the CFMM algorithm calculates the amount of Token B the trader will receive based on the current pool ratios and the amount of Token A they have supplied.

The progress of CFMM over the conventional order book is evident in a number of ways:

i) **Non-Custodial Trading**: DEXs provide users with total control over their assets at all times. Unlike centralised exchanges that require users to deposit cash into custodial accounts, traders retain control of their private keys and assets, minimising the risk of hackers as well as providing a higher level of security;

ii) **Global Accessibility**: DEXs are accessible to anybody with an internet connection, allowing anyone from all over the world to trade digital assets without regard for territorial restrictions or regulatory processes;

iii) **Lower Counterparty Risk**: DEXs reduce counterparty risk by eliminating middlemen. Traders interact with the smart contract directly, removing the need for them to commit their funds to a centralised entity.

In the Ethereum ecosystem, decentralised exchanges offer customers a wide range of capabilities and benefits. They allow users to trade, utilise DeFi services and engage in the increasingly decentralised financial ecosystem in a trustless and safe manner.

The next section of this article will examine four DEXs on the Ethereum blockchain. We will also explore one of the most severe risks that liquidity providers (LPs) face when providing liquidity to a DEX. This risk assessment will provide us with a clear perspective on the shape of the Value Function, the price range in which the LP liquidity is distributed, the Token price definition by the Defi protocol, and the relationship between the Token price and the pool reserve (and the Token weight in the Balancer liquidity pool) in a particular pool will completely determine how any LP will be affected by the Impermanent Loss.

4 Impermanent Loss in Decentralized Exchanges: A Comparative Analysis of the Uniswap and the Balancer Protocol

The main purpose of the demonstrations that follow is that of deriving a deterministic model which when provided with the respective inputs, returns the estimated impermanent loss a Liquidity provider will be subjected to under the circumstances predefined by the LP for the Balancer protocol, Uniswap V2, V3 and Curve Finance.

4.1 Balancer Protocol

Decentralized exchanges, specifically automated market makers (AMMs), play a crucial role in the DeFi ecosystem. Due to their significance, it is essential to examine the risks associated with these products. To compare the expected impermanent loss among major DEXes, we found there was not yet an explicit proof of the impermanent loss formula for Balancer. As a result, this blog post aims to fill that gap by deriving the formula outlined in *Fernando Martinelli's* medium page, which offers a well-presented intuition.

As a quick overview, Balancer is a decentralized exchange that allows users to trade Ethereum-based tokens in a trustless environment. It uses smart contracts to enable users to trade $ERC-20$ tokens, creating liquidity pools for any sequence of tokens. The platform uses a unique algorithm to automatically adjust the token price based on the executed trades (see [10]).

Impermanent Loss: Liquidity pools are often used by traders to exchange tokens in a decentralized manner. However, when liquidity providers (LPs) deposit their tokens into the pool, they may face an opportunity cost. This cost, known as "impermanent loss" (IL), is the measure of the difference in value between holding tokens directly versus indirectly in the pool. The value of tokens within the pool fluctuates over time, resulting in a reduction in the aggregate value of the LPs' investments. Therefore, LPs may experience a loss in value when withdrawing their tokens from the pool (see [14]).

For simplicity's sake, assume there are no deposits or withdrawals in order for us to have a constant value function, K. Notice that, according to the Balancer Whitepaper (see [10]), the spot price (SP) or marginal rate of substitution (MRS) of an input token with respect to the output token was proven to be:

$$\mathrm{SP}_{\mathrm{in}\to\mathrm{out}}(t) = \frac{x_{\mathrm{in}}(t)/\omega_{\mathrm{in}}}{x_{\mathrm{out}}(t)/\omega_{\mathrm{out}}} \ .$$

Where $x_{\mathrm{in}}(t)$ represents the reserve of the token that is being sold at time t, ω_{in} is the correspondent weight in the pool, $x_{\mathrm{out}}(t)$ is the reserve of the token that is being bought at time t, and ω_{out} denotes the respective weight. It is important to emphasize that the weight of each token in any Balancer pool is strictly between zero and one, and when summed, the weights of all tokens add up to unity.

For the sake of simplicity, let us express the price of any token as its current value in dollars per token, which gives us the flexibility to define the price of any i-th token whose reserve is x_{i} as a function of time such as:

$$\begin{aligned} p_i : \mathbb{R}_0^+ &\to \mathbb{R}^+ \\ t &\mapsto p_{\mathrm{i}}(t) \quad [\$/\mathrm{Token_i}] \ . \end{aligned}$$

When a trader wants to buy token i, they can do so directly using an acceptable currency through an exchange, or they can first buy token j and then use it to acquire token i using the pool. To this end, inline with our definition of the spot price and as stated in the Balancer Whitepaper, the price of token i may be expressed as:

$$p_{\mathrm{i}} = p_{\mathrm{j}}\, \mathrm{SP}_{\mathrm{j}\to\mathrm{i}} \, .$$

In the previous price relation, the following property holds true for any two tokens i and j in any Balancer liquidity pool $p_{\mathrm{i}}(t)x_{\mathrm{i}}/\omega_{\mathrm{i}} = p_{\mathrm{j}}(t)x_{\mathrm{j}}/\omega_{\mathrm{j}}$.

Example 1. *As an example, consider that an LP adds BTC, ETH, and BAL as liquidity into a Balancer pool, with weights of* 50%, 30%, *and* 20%, *respectively. The initial prices of BTC, ETH, and BAL are* $p_{\mathrm{BTC}}(t_0) = 50.000$, $p_{\mathrm{ETH}}(t_0) = 3.000$ *and* $p_{\mathrm{BAL}}(t_0) = 20$ *respectively at time* $t = t_0$, *in units of* \$/Token.

At time $t > t_0$ *admit there was a price change such that the current price of BTC, ETH, and BAL in dollars terms are now given by* $p_{\mathrm{BTC}}(t) = 55.000$, $p_{\mathrm{ETH}}(t) = 2.500$, *and* $p_{\mathrm{BAL}}(t) = 25$. *If we consider the following price ratios:*

$$\Delta_{\mathrm{BTC}} = p_{\mathrm{BTC}}(t)/p_{\mathrm{BTC}}(t_0) = 1,1; \quad \Delta_{\mathrm{ETH}} = p_{\mathrm{ETH}}(t)/p_{\mathrm{ETH}}(t_0) \approx 0,833$$
$$\textit{and} \quad \Delta_{\mathrm{BAL}} = p_{\mathrm{BAL}}(t)/p_{\mathrm{BAL}}(t_0) = 1,25\,.$$

By using these inputs into the formula provided by none other than the co-founder of the Balancer protocol Fernando Martinelli himself on his medium page, we can say that the IL this LP will be subjected to at time $t > t_0$ *if they choose to provide liquidity in this hypothetical Balancer pool will be*

$$\mathrm{IL} = \frac{\Delta_{\mathrm{BTC}}^{0,5} \cdot \Delta_{\mathrm{ETH}}^{0,3} \cdot \Delta_{\mathrm{BAL}}^{0,2}}{0,5\Delta_{\mathrm{BTC}} + 0,3\Delta_{\mathrm{ETH}} + 0,2\Delta_{\mathrm{BAL}}} - 1 \approx -1,1\%\,.$$

So in particular, this liquidity provider would have suffered an impermanent loss or opportunity cost of about 1,1% *of his initial capital.*

The impermanent loss formula: In this framework, we want to show that if we consider a liquidity pool containing multiple tokens. Considering the token indexing set I, the impermanent loss, IL, can be given by

$$\mathrm{IL} = \frac{\prod_{\mathrm{i}\in I} \Delta_{\mathrm{i}}^{\omega_{\mathrm{i}}}}{\sum_{\mathrm{i}\in I} \Delta_{\mathrm{i}}\omega_{\mathrm{i}}} - 1\,,$$

where ω_{i} and Δ_{i} represent respectively the pool weight of token i and the associated future and initial price ratio.

Proof for the two token case: We maintain our position by demonstrating that the formula holds in the two-token scenario. We do this to identify all of the phases of the general proof in a less complicated environment.

In this setting, consider that there only two tokens, x and y. The LP invested in the pool by providing assets of both tokens for the pool reserves, which are represented by $x(t)$ and $y(t)$ at the instant t, for x and y, respectively. By defining $0 < \phi_{\mathrm{x}}, \phi_{\mathrm{y}} < 1$ such that $\phi_{\mathrm{x}} + \phi_{\mathrm{y}} = 1$, as the fractions of the pool reserves provided by the LP, the total amounts of x and y in the pool can now given by $x\phi_{\mathrm{x}}$ and $y\phi_{\mathrm{y}}$, correspondingly. Note that, in practice, the LP does not need to provide tokens in the the exact ratio present in the pool.

Let us operate in the case of no deposits or withdrawals, so we can assume that the Balancer pool function, $\hat{K}$, is constant, yielding

$$\hat{K} = [x(t)\phi_{\mathrm{x}}]^{\omega_{\mathrm{x}}} \ [y(t)\phi_{\mathrm{y}}]^{\omega_{\mathrm{y}}} \ ,$$

where ω_{x}, and ω_{y} represent the weight, i.e. the value-proportion of tokens x and y, respectively, in this particular pool. Recall also, that because this is a Balancer weighted pool as previously stated we know that $0 < \omega_{\mathrm{x}}, \omega_{\mathrm{y}} < 1$ and $\omega_{\mathrm{x}} + \omega_{\mathrm{y}} = 1$. Furthermore, by defining

$$K = \phi_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \, \phi_{\mathrm{y}}^{-\omega_{\mathrm{y}}} \, \hat{K}$$

as the pool-invariant associated to the LP, one has

$$K = x(t)^{\omega_{\mathrm{x}}} \, y(t)^{\omega_{\mathrm{y}}} \, . \tag{1}$$

Because both $x(t), y(t)$ denote token quantities in the pool, we naturally have $x(t), y(t) > 0$ for all $t \geq t_0$. As we previously mentioned, the condition $x(t)\, p_{\mathrm{x}}(t)/\omega_{\mathrm{x}} = y(t)\, p_{\mathrm{y}}(t)/\omega_{\mathrm{y}}$ holds, where $p_{\mathrm{x}}(t), p_{\mathrm{y}}(t)$ represent respectively, the token x and y prices (in dollars per token) at any time $t \geq t_0$. Hence we have that

$$p_{\mathrm{x}}(t) = \frac{y(t)}{x(t)} \frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} p_{\mathrm{y}}(t), \quad \text{and} \quad p_{\mathrm{y}}(t) = \frac{x(t)}{y(t)} \frac{\omega_{\mathrm{y}}}{\omega_{\mathrm{x}}} p_{\mathrm{x}}(t) \, . \tag{2}$$

If we substitute $y(t)$ in the equation 2 using the relation 1, it follows that:

$$\begin{aligned} & p_{\mathrm{x}}(t) = K^{\frac{1}{\omega_{\mathrm{y}}}} \, x(t)^{-(1+\frac{\omega_{\mathrm{y}}}{\omega_{\mathrm{x}}})} \frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} p_{\mathrm{y}} y(t) \iff x(t) = \left(K^{\frac{1}{\omega_{\mathrm{y}}}} \, \frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{x}}(t)} \right)^{\frac{1}{1+\frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}}}} \iff \\ & \iff x(t) = \left(K^{\frac{1}{\omega_{\mathrm{y}}}} \, \frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{x}}(t)} \right)^{\omega_{\mathrm{y}}}, \quad \text{as} \quad \omega_{\mathrm{x}} + \omega_{\mathrm{y}} = 1 \implies \frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} = \frac{1}{\omega_{\mathrm{y}}} - 1 \\ & \iff x(t) = K \left(\frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{x}}(t)} \right)^{\omega_{\mathrm{y}}} . \end{aligned}$$

And, analogously, we can conclude that:

$$y(t) = K \left(\frac{\omega_{\mathrm{y}}}{\omega_{\mathrm{x}}} \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{y}}(t)} \right)^{\omega_{\mathrm{x}}} .$$

Hence, the value (in dollars) at time t the LP invested, i.e., $V_{\mathrm{invest}}(t)$ for $t \geq t_0$, will be given by

$$\begin{aligned}
V_{\mathrm{invest}}(t) &= x(t)\, p_{\mathrm{x}}(t) + y(t)\, p_{\mathrm{y}}(t) \\
&= K \left(\frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{x}}(t)} \right)^{\omega_{\mathrm{y}}} p_{\mathrm{x}}(t) + K \left(\frac{\omega_{\mathrm{y}}}{\omega_{\mathrm{x}}} \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{y}}(t)} \right)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t) \\
&= K \left[\omega_{\mathrm{x}}^{\omega_{\mathrm{y}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\, p_{\mathrm{x}}(t)^{1-\omega_{\mathrm{y}}} p_{\mathrm{y}}(t)^{\omega_{\mathrm{y}}} + \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{1-\omega_{\mathrm{y}}}\, p_{\mathrm{x}}(t)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t)^{1-\omega_{\mathrm{x}}} \right] \\
&= K\, \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\, p_{\mathrm{x}}(t)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t)^{\omega_{\mathrm{y}}} \left(\omega_{\mathrm{x}} + \omega_{\mathrm{y}} \right) = K\, \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\, p_{\mathrm{x}}(t)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t)^{\omega_{\mathrm{y}}} .
\end{aligned}$$

Had the LP held their tokens instead, the asset quantities would have remained constant, and we can derive their value at time $t > t_0$ which will be denoted by $V_{\mathrm{hold}}(t)$. Notice that we may substitute the initial (and constant) quantities of the LP's reserves by the pool's relation, because in the initial instant there is, by definition, no price variation - this is will be useful later for direct comparison between the expressions. It follows:

$$\begin{aligned}
V_{\mathrm{hold}}(t) &= x(t_0)\, p_{\mathrm{x}}(t) + y(t_0)\, p_{\mathrm{y}}(t) = \\
&= K \left(\frac{\omega_{\mathrm{x}}}{\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t_0)}{p_{\mathrm{x}}(t_0)} \right)^{\omega_{\mathrm{y}}} p_{\mathrm{x}}(t) + K \left(\frac{\omega_{\mathrm{y}}}{\omega_{\mathrm{x}}} \frac{p_{\mathrm{x}}(t_0)}{p_{\mathrm{y}}(t_0)} \right)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t) = \\
&= K \left(\omega_{\mathrm{x}}^{\omega_{\mathrm{y}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}} \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{x}}(t_0)} p_{\mathrm{x}}(t_0)^{1-\omega_{\mathrm{y}}} p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}} + \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{\omega_{\mathrm{x}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{y}}(t_0)} p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t_0)^{1-\omega_{\mathrm{x}}} \right) = \\
&= K \left(\omega_{\mathrm{x}}^{1-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}} \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{x}}(t_0)} p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}} + \omega_{\mathrm{y}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{1-\omega_{\mathrm{y}}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{y}}(t_0)} p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}} \right) = \\
&= K\, \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}} p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}} \left(\omega_{\mathrm{x}} \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{x}}(t_0)} + \omega_{\mathrm{y}} \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{y}}(t_0)} \right) .
\end{aligned}$$

If we consider the following notations for sake of simplicity:

$$\Delta_{\mathrm{x}}(t_0, t) = \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{x}}(t_0)}, \quad \text{and} \quad \Delta_{\mathrm{y}}(t_0, t) = \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{y}}(t_0)} ,$$

then the previous result becomes:

$$V_{\mathrm{hold}}(t) = K\, \omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}} \omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}} p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}} p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}} \left(\omega_{\mathrm{x}} \Delta_{\mathrm{x}} + \omega_{\mathrm{y}} \Delta_{\mathrm{y}} \right) ,$$

Conjugating the expressions for $V_{\mathrm{invest}}(t)$ and $V_{\mathrm{hold}}(t)$, and taking into account the definition of impermanent loss (IL), we may conclude that the IL, that the LP will be subjected to, at the maturity time $t > t_0$, is given by:

$$\begin{aligned}\mathrm{IL} = \frac{V_{\text{invest}}(t) - V_{\text{hold}}(t)}{V_{\text{hold}}(t)} &= \frac{K\,\omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}}\,\omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\,p_{\mathrm{x}}(t)^{\omega_{\mathrm{x}}}\,p_{\mathrm{y}}(t)^{\omega_{\mathrm{y}}}}{K\,\omega_{\mathrm{x}}^{-\omega_{\mathrm{x}}}\,\omega_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\,p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}}\,p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}}\,(\omega_{\mathrm{x}}\Delta_{\mathrm{x}} + \omega_{\mathrm{y}}\Delta_{\mathrm{y}})} - 1 = \\ = \frac{p_{\mathrm{x}}(t)^{\omega_{\mathrm{x}}}\,p_{\mathrm{y}}(t)^{\omega_{\mathrm{y}}}}{p_{\mathrm{x}}(t_0)^{\omega_{\mathrm{x}}}\,p_{\mathrm{y}}(t_0)^{\omega_{\mathrm{y}}}\,(\omega_{\mathrm{x}}\Delta_{\mathrm{x}} + \omega_{\mathrm{y}}\Delta_{\mathrm{y}})} - 1 &= \frac{\Delta_{\mathrm{x}}(t_0,t)^{\omega_{\mathrm{x}}}\Delta_{\mathrm{y}}(t_0,t)^{\omega_{\mathrm{y}}}}{\omega_{\mathrm{x}}\Delta_{\mathrm{x}}(t_0,t) + \omega_{\mathrm{y}}\Delta_{\mathrm{y}}(t_0,t)} - 1\,.\end{aligned}$$

Now that we have built our intuition by proving the result for the simplest case, when we have a liquidity pool composed of two ERC-tokens, let us prove that this result holds when consider a pool made of an abstract finite number of tokens.

Proof for the multiple token case: The current reserve of token i in the multiple-tokens Balancer weighted pool is denoted as $x_{\mathrm{i}}(t)$, for any $\mathrm{i} \in I$, where I is the indexing set for the tokens in the pool.

For example, one could have: $I = \{\mathrm{BTC}, \mathrm{ETH}, \mathrm{BAL}\}$. Notice that even for a liquidity pool with multiple tokens, we can still assert the base premise we have been using thus far still holds true, i.e. for each $\mathrm{i}, \mathrm{j} \in I$,

$$\frac{x_{\mathrm{i}}(t)}{\omega_{\mathrm{i}}}\,p_{\mathrm{i}}(t) = \frac{x_{\mathrm{j}}(t)}{\omega_{j}}\,p_{\mathrm{j}}(t) \iff x_{\mathrm{j}}(t) = x_{\mathrm{i}}(t)\frac{\omega_{\mathrm{j}}}{\omega_{\mathrm{i}}}\frac{p_{\mathrm{i}}(t)}{p_{\mathrm{j}}(t)}. \tag{3}$$

similarly to the case of 2-tokens, we can isolate $x_{\mathrm{i}}(t)$ from the invariant associated with the LP amount of tokens, which yields

$$x_{\mathrm{i}}(t) = K^{\frac{1}{\omega_{\mathrm{i}}}} \prod_{\mathrm{j} \in I \setminus \{\mathrm{i}\}} x_{\mathrm{j}}(t)^{-\omega_{\mathrm{j}}/\omega_{\mathrm{i}}}. \tag{4}$$

It is important to emphasize that the weights sum up to one, which leads to

$$\sum_{\mathrm{s} \in I \setminus \{\mathrm{i}\}} \omega_{\mathrm{s}} = 1 - \omega_{\mathrm{i}} \quad \text{for each} \quad \mathrm{i} \in I.$$

By replacing the reserve of token j in equation (4) with its correspondent expression from

(3), we obtain

$$\begin{aligned}
&x_{\mathrm{i}}(t) = K^{\frac{1}{\omega_{\mathrm{i}}}} \prod_{\mathrm{j}\in I\setminus\{\mathrm{i}\}} \left(x_{\mathrm{i}}(t) \frac{\omega_{\mathrm{j}}}{\omega_{\mathrm{i}}} \frac{p_{\mathrm{i}}(t)}{p_{\mathrm{j}}(t)} \right)^{-\omega_{\mathrm{j}}/\omega_{\mathrm{i}}} \iff \\
\iff &x_{\mathrm{i}}(t) = K^{\frac{1}{\omega_{\mathrm{i}}}} \left(\frac{x_{\mathrm{i}}(t)\, p_{\mathrm{i}}(t)}{\omega_{\mathrm{i}}} \right)^{-\sum\limits_{s\in I\setminus\{\mathrm{i}\}} \omega_s/\omega_{\mathrm{i}}} \prod_{\mathrm{j}\in I\setminus\{\mathrm{i}\}} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}/\omega_{\mathrm{i}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}/\omega_{\mathrm{i}}} \iff \\
\iff &x_{\mathrm{i}}(t)^{1+\sum\limits_{\mathrm{s}\in I\setminus\{\mathrm{i}\}} \omega_{\mathrm{s}}/\omega_{\mathrm{i}}} = K^{\frac{1}{\omega_{\mathrm{i}}}} \left(\frac{p_{\mathrm{i}}(t)}{\omega_{\mathrm{i}}} \right)^{1-\frac{1}{\omega_{\mathrm{i}}}} \prod_{\mathrm{j}\in I\setminus\{\mathrm{i}\}} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}/\omega_{\mathrm{i}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}/\omega_{\mathrm{i}}} \iff \\
\iff &x_{\mathrm{i}}(t)^{\frac{1}{\omega_{\mathrm{i}}}} = K^{\frac{1}{\omega_{\mathrm{i}}}} p_{\mathrm{i}}(t)^{1-\frac{1}{\omega_{\mathrm{i}}}} \omega_{\mathrm{i}}^{-1+\frac{1}{\omega_{\mathrm{i}}}} \prod_{\mathrm{j}\in I\setminus\{\mathrm{i}\}} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}/\omega_{\mathrm{i}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}/\omega_{\mathrm{i}}} \iff \\
\iff &x_{\mathrm{i}}(t) = K \frac{\omega_{\mathrm{i}}}{p_{\mathrm{i}}(t)} \prod_{\mathrm{j}\in I} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}}, \quad \text{for} \quad t \geq t_0.
\end{aligned}$$

Therefore the fraction of the Pool value that belongs to the LP at time $t > t_0$ in dollar terms is

$$\begin{aligned}
V_{\text{invest}}(t) &= \sum_{\mathrm{i}\in I} p_{\mathrm{i}}(t) x_{\mathrm{i}}(t) = K \sum_{\mathrm{i}\in I} \left(p_{\mathrm{i}}(t) \frac{\omega_{\mathrm{i}}}{p_{\mathrm{i}}(t)} \prod_{\mathrm{j}\in I} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}} \right) \\
&= K \left(\sum_{\mathrm{i}\in I} \omega_{\mathrm{i}} \right) \prod_{\mathrm{j}\in I} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}} = K \prod_{\mathrm{j}\in I} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}} \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}}.
\end{aligned}$$

Again, had the LP held his or her tokens instead of providing the liquidity with them to this pool, the quantities of each would have remained the same from time $t = t_0$, expressly $x_{\mathrm{i}}(t) = x_{\mathrm{i}}(t_0)$, for $t \geq t_0$. Hence, the hold value in dollars is given by

$$\begin{aligned}
V_{\text{hold}}(t) &= \sum_{\mathrm{i}\in I} p_{\mathrm{i}}(t)\, x_{\mathrm{i}}(t_0) = K \sum_{\mathrm{i}\in I} \left(p_{\mathrm{i}}(t) \frac{\omega_{\mathrm{i}}}{p_{\mathrm{i}}(t_0)} \prod_{\mathrm{j}\in I} p_{\mathrm{j}}(t_0)^{\omega_{\mathrm{j}}}\, \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}} \right) = \\
&= K \sum_{\mathrm{i}\in I} \left(\omega_{\mathrm{i}} \frac{p_{\mathrm{i}}(t)}{p_{\mathrm{i}}(t_0)} \right) \prod_{\mathrm{j}\in I} \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}} p_{\mathrm{j}}(t_0)^{\omega_{\mathrm{j}}}.
\end{aligned}$$

Having the definition of impermanent loss in mind, we may conclude that the LP will face impermanent loss at time $t \geq t_0$, which will be given by

$$\begin{aligned}
\mathrm{IL}(t) &= \frac{V_{\text{invest}}(t) - V_{\text{hold}}(t)}{V_{\text{hold}}(t)} = \frac{K \prod\limits_{\mathrm{j}\in I} p_{\mathrm{j}}(t)^{\omega_{\mathrm{j}}} \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}}}{K \sum\limits_{\mathrm{i}\in I} \left(\omega_{\mathrm{i}} \frac{p_{\mathrm{i}}(t)}{p_{\mathrm{i}}(t_0)} \right) \prod\limits_{\mathrm{j}\in I} \omega_{\mathrm{j}}^{-\omega_{\mathrm{j}}} p_{\mathrm{j}}(t_0)^{\omega_{\mathrm{j}}}} - 1 = \\
&= \frac{\prod\limits_{\mathrm{j}\in I} \left(\frac{p_{\mathrm{j}}(t)}{p_{\mathrm{j}}(t_0)} \right)^{\omega_{\mathrm{j}}}}{\sum\limits_{\mathrm{i}\in I} \omega_{\mathrm{i}} \frac{p_{\mathrm{i}}(t)}{p_{\mathrm{i}}(t_0)}} - 1 = \frac{\prod\limits_{\mathrm{j}\in I} \Delta_{\mathrm{j}}(t_0, t)^{\omega_{\mathrm{j}}}}{\sum\limits_{\mathrm{i}\in I} \omega_{\mathrm{i}} \Delta_{\mathrm{i}}(t_0, t)} - 1,
\end{aligned}$$

where we are considering $\Delta_{\mathrm{i}}(t_0, t) = p_{\mathrm{i}}(t)/p_{\mathrm{i}}(t_0)$ for each $\mathrm{i} \in I$.

5 Uniswap Protocol

Uniswap is a DeFi decentralized exchange (DEx) pioneer. It allows users to trade cryptocurrencies without the use of traditional middlemen and provides liquidity via automated smart contracts. Its user-friendly interface and liquidity pool strategy have made it a popular choice in the field of blockchain and cryptocurrencies for decentralized token exchanges and liquidity provision.

5.1 *Version Two (V2)*

Uniswap V2 is a decentralized Ethereum exchange for trading $ERC-20$ tokens. It improves trading efficacy, flexibility, liquidity, and reliability over the original Uniswap protocol. Notable improvements include liquidity pools for any pair of $ERC-20$ compliant tokens, in which users contribute equal dollar amounts of two tokens and thereby become liquidity providers (LPs). Fees are paid to LPs based on the liquidity they offer. For its automated market maker (AMM) method, Uniswap V2 uses a constant product formula, providing rapid transactions and changeable token values depending on supply and demand. Overall, Uniswap V2 provides a liquid, versatile, and cost-effective decentralized trading environment that is popular among DeFi users and serves as the foundation for Ethereum decentralized exchanges (see [1]).

Progressing in our discussion concerning the Impermanent Loss (IL) one of the major protocols in DeFi, Uniswap V2, the formula for the spot price of an input token with respect to the output token (the price for infinitesimally small trades) can be expressed as follows:

$$\mathrm{SP}_{\mathrm{in}\to\mathrm{out}}(t)=\frac{x_{\mathrm{in}}(t)}{x_{\mathrm{out}}(t)} \tag{5}$$

where $x_{\mathrm{in}}(t)$ denotes the reserve of the token being sold at time $t \geq t_0$, and $x_{\mathrm{out}}(t)$ represents the reserve of the token being bought (see reference [?]). It is important to notice that in each Uniswap V2 pool, the liquidity is distributed uniformly along the curve defined by the following equation:

$$\hat{K}_{\mathrm{V2}}=x(t)\,y(t) \tag{6}$$

where $x(t)$ and $y(t)$ represents the current total amount of tokens x and y, respectively, in that particular liquidity pool. This curve, usually named Value Function, will play a crucial role in each model we intend to derive the IL formula in our analysis case of Uniswap V2.

Along our discussion unless otherwise stated, we will assume there are no deposits nor withdrawals in order for us to have a constant value function, $\hat{K} = \hat{K}_{\mathrm{V2}}$. Furthermore, we will introduce the price of each token as its current value in dollars per token, which is to say that if we consider $x_{\mathrm{A}}(t), t \geq t_0$ the current total reserve of token A in a given pool, then:

$$t \mapsto p_{\mathrm{A}}(t) \quad [\$/\mathrm{Token}_{\mathrm{A}}]$$

will represent the price of token A at time $t \geq t_0$ in the units of dollar per token. The following equation gives birth to our essential premises:

$$p_{\mathrm{x}}(t) \cdot x(t) = p_{\mathrm{y}}(t) \cdot y(t) \tag{7}$$

Notice that this property holds true for any two $ERC-20$ tokens x and y, in any Uniswap V2 liquidity pool, by the same argument we make in the Balancer IL formula derivation in the previous section.

Let us assume the LP invested in the pool by providing assets of both tokens for a given pool made of two given $ERC-20$ tokens, x and y, whose total tokens reserves at instant $t \geq t_0$ are denoted by $x(t)$ and $y(t)$. Taking $0 < \phi_{\mathrm{x}}, \phi_{\mathrm{y}} < 1$ such that $\phi_{\mathrm{x}} + \phi_{\mathrm{y}} = 1$ as the fractions of the pool reserves provided by the LP, the LP's total share amounts of both tokens in the pool can now be given by $x(t)\,\phi_{\mathrm{x}}$ and $y(t)\,\phi_{\mathrm{y}}$, respectively which leads us to our adapted formulation for the value function

$$K = \phi_{\mathrm{x}}\,\phi_{\mathrm{y}}\hat{K} = \big[\phi_{\mathrm{x}}\,x(t)\big]\,\big[\phi_{\mathrm{y}}\,y(t)\big]$$

Notice that, in the particular case when the token weights in a Balancer (two tokens) liquidity pool are the same, it signifies that the pool strives to maintain a balanced distribution of the two tokens similar to assuring a constant ratio between token reserves, which corresponds to an equivalent constant product formula used by Uniswap V2 expressed in equation 6.

Both Uniswap V2 and the equal-weighted Balancer (two tokens) liquidity pool comply to the token reserve's equivalent constant product value function. This fact ensures that in this particular scenario Uniswap V2 and Balancer are managed the same way by an equivalent value function, and in our setting while exchanges take place in the Balancer decentralized Exchange, the product of the reserves of the two tokens in a given pool is governed by

$$\sqrt{L} = x(t)^{\omega_{\mathrm{x}}}\, y(t)^{\omega_{\mathrm{y}}}$$

where we are introducing $L = \sqrt{\phi_{\mathrm{x}}^{-\omega_{\mathrm{x}}}\, \phi_{\mathrm{y}}^{-\omega_{\mathrm{y}}}\, K}$ as the pool-invariant associated to the LP position in the pool, and $\omega_{\mathrm{x}} = \omega_{\mathrm{y}} = 1/2$ represents the weight of the respective token in Balancer (two tokens) liquidity pool which we argue that its value function are mathematically speaking the same, in this case only. As proven in the beginning of our discussion, the IL formula for the Balancer protocol in these circumstances taking into consideration its correspondent value function is given by

$$\mathrm{IL}(t) = \frac{\Delta_{\mathrm{x}}^{\omega_{\mathrm{x}}} \Delta_{\mathrm{y}}^{\omega_{\mathrm{y}}}}{\omega_{\mathrm{x}}\Delta_{\mathrm{x}} + \omega_{\mathrm{y}}\Delta_{\mathrm{y}}} - 1 = 2\,\frac{\sqrt{\Delta_{\mathrm{x}}\,\Delta_{\mathrm{y}}}}{\Delta_{\mathrm{x}} + \Delta_{\mathrm{y}}} - 1\,.$$

Where we are considering our already acquainted notation

$$\Delta_{\mathrm{x}} = \Delta_{\mathrm{x}}(t_0, t) = \frac{p_{\mathrm{x}}(t)}{p_{\mathrm{x}}(t_0)} \qquad \text{and} \qquad \Delta_{\mathrm{y}} = \Delta_{\mathrm{y}}(t_0, t) = \frac{p_{\mathrm{y}}(t)}{p_{\mathrm{y}}(t_0)}\,.$$

As we previously argued, due to the fact that in this particular case where the two tokens in the (two token) Balancer liquidity pool are equally weighted, Balancer and Uniswap V2 pools are managed by two equivalent value functions. In this regard, because the IL formula for the Balancer two tokens liquidity pools is as provided so must be the IL formula for the Uniswap V2. This concludes our proof of the IL formula for Uniswap V2 liquidity pool.

5.2 *Version Three (V3)*

In straightforward terms, Uniswap V3 allows liquidity providers to provide liquidity selectively within defined price ranges for better capital efficiency and risk management. The provision of several price levels helps providers align their rates with their risk profiles. These distinguishing features empower liquidity providers, solidifying Uniswap V3's position as one of the best decentralized exchanges.

Notice that in earlier versions of Uniswap, the liquidity of each pool is distributed uniformly along the Constant Product curve which also means that all the revenue from the trading fee has to be distributed to all the Liquidity Providers in the respective pool in proportion to their share token in the pool. With the introduction of the Concentrated

Liquidity feature the LPs has are given the flexibility of defining the price range as they prefer which we will denote abstractly as $[p_{\mathrm{i}}, p_{\mathrm{s}}]$ as of $p_{infimum-range}, p_{supremum-range}$. Due to the existence of a smaller number of market participants providing liquidity in each price range, revenue from the trading fee is distributed to less LPs and the liquidity tends to be more concentrated in the vicinity of the current price. Although the real reserve in each pool of the protocol is fragmented between price ranges, within the pool the token price is still calculated as in equation 5 considering the total reserve in the pool usually called Virtual Reserves (see [2]).

Impermanent Loss Formula for Uniswap V3

Before we start our demonstration, consider the definition of the following function which will play a significant rule in our proof

$$f(t) = \sqrt{\left(\frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}} - p_{\mathrm{y}}(t)\sqrt{p_{\mathrm{i}}}\right)^2 + 4p_{\mathrm{x}}(t)p_{\mathrm{y}}(t)} - \frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}} - p_{\mathrm{y}}(t)\sqrt{p_{\mathrm{i}}} \tag{8}$$

We are going to provide a proof of the IL formula for any Uniswap V3 liquidity pool which is given by

$$\mathrm{IL}(t) = \frac{2}{\Delta_{\mathrm{x}} + \Delta_{\mathrm{y}}} \, \frac{f(t)}{f(t_0)} - 1\,.$$

Where we are considering a pool of token x and y whose virtual reserves are represented by $x(t)$ and $y(t)$ at the instant $t \geq t_0$, respectively, and the function $f(t)$ defined above.

Proof Assuming the settings absolutely analogous as in the proof we provided in the Balancer two token pool case, let us also consider that there are no deposits or withdrawals, so we can assume that the pool Value Function, $\hat{K} = \hat{K}_{V3}$, is constant, i.e.

$$\hat{K} = \left(\frac{x(t)}{\phi_{\mathrm{x}}}\right)\left(\frac{y(t)}{\phi_{\mathrm{y}}}\right)\,,$$

where $0 < \phi_{\mathrm{x}}, \phi_{\mathrm{y}} < 1$ denotes as the fractions of the pool reserves provided by the LP, and $\phi_{\mathrm{x}} + \phi_{\mathrm{y}} = 1$. And we can further define $K_{\mathrm{V3}} = \phi_{\mathrm{x}}\,\phi_{\mathrm{y}}\,\hat{K}$ as the pool-invariant associated to the LP, which simplifies the previous equation, yielding $K_{\mathrm{V3}} = x(t)\,y(t)$ as the value function for the virtual reserve.

Now consider $K = K_{\mathrm{V3}}$, given that the liquidity in the pool is fragmented due to the concentrated liquidity nature, the Value Function that governs the fraction of the real reserve in the arbitrary price range chosen by the LP which we will denote as $[p_\mathrm{i}, p_\mathrm{s}]$ is given according to the expression 2.) in the Uniswap V3 whitepape (see [2]) as

$$K = \left(x(t) + \sqrt{\frac{K}{p_\mathrm{s}}}\right)\left(y(t) + \sqrt{Kp_\mathrm{i}}\right)$$

The origin of real reserve invariant formula: The Uniswap V3 real reserve value function formula corresponds to a translation of the constant product invariant express in the Figure 1 of the whitepaper, yielding

$$(x - x_0)(y - y_0) = K(x, y) > 0$$

with $x_0, y_0 < 0$ representing the position of the vertical and horizontal asymptotes, respectively. The translation is of $-x_0$ units to the left and $-y_0$ units downwards, in a x, y cartesian plot. For this invariant K, the marginal rate of substitution from y to x (spot-price) becomes

$$\mathrm{SP}_{\mathrm{y}\to\mathrm{x}} = \frac{\left(\dfrac{\partial K}{\partial x}\right)}{\left(\dfrac{\partial K}{\partial y}\right)} = \frac{y - y_0}{x - x_0} = \frac{K}{(x - x_0)^2} = \frac{(y - y_0)^2}{K} > 0\,.$$

At $x = 0$, the marginal rate of substitution from y to x takes its maximum value, p_s, as follows

$$p_\mathrm{s} = \mathrm{SP}_{\mathrm{y}\to\mathrm{x}}\Big|_{x=0} = \frac{K}{(x - x_0)^2}\Bigg|_{x=0} = \frac{K}{x_0^2} \implies x_0 = -\sqrt{\frac{K}{p_\mathrm{s}}}\,.$$

Which the scenario where all the liquidity is being provided only to the asset which has $y(t)$ as the current pool virtual reserve. At $y = 0$, the marginal rate of substitution from y to x takes its minimum value, p_i, thus

$$p_\mathrm{i} = \mathrm{SP}_{\mathrm{y}\to\mathrm{x}}\Big|_{y=0} = \frac{(y - y_0)^2}{K}\Bigg|_{y=0} = \frac{y_0^2}{K} \implies y_0 = -\sqrt{Kp_\mathrm{i}}\,.$$

Hence, substituting in the values of x_0 and y_0, the Uniswap V3 invariant may be written in terms of the concentrated liquidity price range as follows

$$\left(x + \sqrt{\frac{K}{p_\mathrm{s}}}\right)\left(y + \sqrt{Kp_\mathrm{i}}\right) = K\,. \tag{9}$$

Using the propriety described in equation 7 in conjunction with the one in the equation 9 yields

$$K = \left(x(t) + \sqrt{\frac{K}{p_\mathrm{s}}}\right)\left(x(t)\,\frac{p_\mathrm{x}(t)}{p_\mathrm{y}(t)} + \sqrt{K p_\mathrm{i}}\right) \quad \text{and}$$
$$K = \left(y(t)\,\frac{p_\mathrm{y}(t)}{p_\mathrm{x}(t)} + \sqrt{\frac{K}{p_\mathrm{s}}}\right)\left(y(t) + \sqrt{K p_\mathrm{i}}\right) .$$

Solving the previous two quadratic equations with respect to $x(t)$ and $y(t)$ respectively, and taking the positive solutions, leads to:

$$x(t) = \frac{\sqrt{K}}{2}\frac{f(t)}{p_\mathrm{x}(t)} \quad \text{and} \quad y(t) = \frac{\sqrt{K}}{2}\frac{f(t)}{p_\mathrm{y}(t)}\,, \tag{10}$$

where $f(t)$ is defined in 8. In these circumstances, we may conclude that the LP investment amount in dollars terms at time $t \geq t_0$, is given by

$$V_\mathrm{invest}(t) = x(t)\,p_\mathrm{x}(t) + y(t)\,p_\mathrm{y}(t) = \sqrt{K} f(t)$$

Let's now consider the scenario where the LP did not provide liquidity to the pool. The asset quantities would have remained constant if the LP had maintained their tokens, and we can compute their value at time $t > t_0$. We may substitute the initial amounts of the LP's reserves by his correspondent pool's fractional reserve as there is no price movement in the initial instant, using equation 10 at time $t = t_0$. Hence, we get

$$\begin{aligned} V_\mathrm{hold}(t) &= x(t_0)\,p_\mathrm{x}(t) + y(t_0)\,p_\mathrm{y}(t) \\ &= x(t_0)\,p_\mathrm{x}(t_0)\,\Delta_\mathrm{x} + y(t_0)\,p_\mathrm{y}(t_0)\,\Delta_\mathrm{y} = \frac{\sqrt{K}}{2}\left(\Delta_\mathrm{x} + \Delta_\mathrm{y}\right) f(t_0)\,. \end{aligned}$$

Hence we may conclude that the IL the LP will be subjected to at time $t \geq t_0$ is given by

$$\mathrm{IL}(t) = \frac{V_\mathrm{invest}(t) - V_\mathrm{hold}(t)}{V_\mathrm{hold}(t)} = \frac{2}{\Delta_\mathrm{x} + \Delta_\mathrm{y}}\frac{f(t)}{f(t_0)} - 1\,,$$

which concludes our discussion on Uniswap V3 Decentralized Exchange. Notice that this formula is in accordance with the proof provided in the case of Uniswap V2 which corresponds to the case where the liquidity in the pool is uniformly distributed along the curve described by the virtual value function in the entire $\mathbb{R}_0^+$ space. Because if we consider $p_\mathrm{i} = 0$ and $p_\mathrm{s} \to +\infty$, we obtain the same IL formula as for Uniswap V2.

Proof: Fix the parameter $p_{\mathrm{i}} = 0$. It follows that the limit of the f, as p_{s} tends to infinity, yields

$$\begin{aligned}
&\lim_{p_{\mathrm{s}}\to+\infty} f(t) = \\
&= \lim_{p_{\mathrm{s}}\to+\infty} \sqrt{\left(\frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}} - p_{\mathrm{y}}(t)\sqrt{p_{\mathrm{i}}}\right)^2 + 4p_{\mathrm{x}}(t)p_{\mathrm{y}}(t)} - \frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}} - p_{\mathrm{y}}(t)\sqrt{p_{\mathrm{i}}} = \\
&= \lim_{p_{\mathrm{s}}\to+\infty} \sqrt{\left(\frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}}\right)^2 + 4p_{\mathrm{x}}(t)p_{\mathrm{y}}(t)} - \frac{p_{\mathrm{x}}(t)}{\sqrt{p_{\mathrm{s}}}} = \sqrt{4\,p_{\mathrm{x}}(t)\,p_{\mathrm{y}}(t)} > 0\,.
\end{aligned}$$

therefore, regarding the impermanent loss, notice when we fixed $p_i = 0$ and take the limit with respect to p_s we have

$$\begin{aligned}
\lim_{p_{\mathrm{s}}\to+\infty} \mathrm{IL}_{\mathrm{V3}}(t) &= \lim_{p_{\mathrm{s}}\to+\infty} \frac{2}{\Delta_{\mathrm{x}} + \Delta_{\mathrm{y}}} \frac{f(t)}{f(t_0)} - 1 = \frac{2}{\Delta_{\mathrm{x}} + \Delta_{\mathrm{y}}} \frac{\sqrt{4\,p_{\mathrm{x}}(t)\,p_{\mathrm{y}}(t)}}{\sqrt{4\,p_{\mathrm{x}}(t_0)\,p_{\mathrm{y}}(t_0)}} - 1 = \\
&= 2\,\frac{\sqrt{\Delta_{\mathrm{x}}\,\Delta_{\mathrm{y}}}}{\Delta_{\mathrm{x}} + \Delta_{\mathrm{y}}} - 1 = \mathrm{IL}_{\mathrm{V2}}(t).
\end{aligned}$$

6 Stochastic Modeling of Token Prices

Stochastic Calculus is a branch of Mathematics used to study systems whose evolution over time $X_t, t \geq 0$ can be explained with help of a differential equation which conjugate both a deterministic function and a random source of noise. Usually the dynamics of the system is described as:

$$dX_t = \mu\,(t, X_t)\,dt + \sigma\,(t, X_t)\,dB_t$$

where B_t denotes the source of randomness. As stated in the previous section, to have a projection of token price evolution from the present up until a future instance in time we will use a much more elegant type of Lévy stochastic process known as the Variance Gamma (see [5] and [6]).

6.1 Lévy Process

In our following discussion, unless otherwise stated, we will be working in a probability space $(\Omega, \mathscr{F}, P)$.

Question 1. *But what is a stochastic process? We provide the answer to this question as well as a definition of some classes of stochastic processes in the Appendix B.*

The application of Stochastic processes in the real world is vast. To name a few, Stochastic processes are used in:

i) Physics: to model the behavior of subatomic particles, to describe the interaction between atoms and photons or to model the decay of unstable atomic nuclei;

ii) The aviation: stochastic processes play a crucial role in designing and maintaining aircraft engines. Given the wide range of random events that engines are subject to, such as fluctuations in air pressure and temperature, stochastic processes are employed to model and predict the behaviour of these events. The insights gained from these processes are then utilized to enhance the design of aircraft engines and schedule maintenance and repairs accordingly.

iii) Mathematical Finance: to model volatility, to price complex financial instruments and to model the dynamics of stock price

The Black-Scholes model is a mathematical formula that provides a process for pricing options contracts based on a variety of criteria such as the underlying asset price, time till expiry, volatility, interest rates, and strike price.

The Black-Scholes model is a widely used financial tool that assumes the price of an underlying asset follows a log-normal distribution and that there are no arbitrage opportunities in the market. It was introduced in 1973 by Fischer Black and Myron Scholes for option pricing and risk management. Over time, the model has been improved to account for dividend-paying assets, time-dependent volatility and drift, and consumption at a specific rate. However, the Black-Scholes model has some significant limitations. One of these is that the price dynamics of the risky asset are given as a continuous function of a diffusion process called Geometric Brownian Motion, which fails to capture the fact that securities volatility is stochastic and asset prices have frequent jumps occurring at random times.

Alternatively, a model driven by Lévy processes is by nature a significantly more reliable and robust approach to modeling securities price dynamics in financial markets. The Lévy stochastic process is defined as follow:

Definition 1. *An adapted stochastic process $L_t, t \in [0,T]$ is a Lévy process if $L_0 = 0$ a.s. and*

(1) L has independent and stationary increments;

(2) L is stochastically continuous, i.e.,

$$\lim_{s \to t} P\left\{|L_t - L_s| > \delta\right\} = 0 \quad \forall t \in [0,T], \forall \delta > 0,$$

A Lévy Stochastic process is composed of a deterministic function, a diffusion random variable as well as a pure jump random variable and it is completely described by the Lévy triplet (γ, σ, ν) . Usually, the parameter γ represents the drift of the process, σ denotes the diffusion component, and ν represents the Lévy measure associated with the jumps term (see [5] and [6]).

The following characterization provides us a way to write any Lévy stochastic process.

Theorem 1. *(Lévy-Itô decomposition): For each Lévy process X, there is a constant $b \in \mathbb{R}$, a Brownian motion B, and an independent Poisson random measure N such that:*

$$X_t = bt + B_t + \int_{|x|<\epsilon} x\tilde{N}(t,dx) + \int_{|x|\geq\epsilon} xN(t,dx)$$

for any arbitrarily small $\epsilon > 0$.

Corollary 1. *(Lévy-Khintchine formula) Consider a Lévy triplet (γ, σ, ν) of the Lévy process X. Then,*

$$\Phi_X(u) = e^{t\eta(u)}, \quad \forall u \in \mathbb{R}$$

where the Lévy exponent η, is defined as:

$$\eta(u) = ibu - \frac{1}{2}\sigma^2 u^2 + \int_{\mathbb{R}-0} \left(e^{iux} - 1 - iux\mathbf{1}_{|x|<1}(x)\right)\nu(dx)$$

To calculate the average rate of return for a token that will be used to define the price process, we can utilize a stochastic process introduced in 1985 by John C. Cox, and Jonathan E. Ingersoll and Stephen A. Ross. This process is an extension of the Vasicek model and is defined as follows:

Definition 2. *Consider a mean-reverting positive stochastic process* $\{y_t, t \geq 0\}$ *and the standard Brownian motion* $\{B_t, t \geq 0\}$*. The CIR process have the following dynamic:*

$$dy_t = \kappa \left(\mu - y_t\right) dt + \alpha\sqrt{y_t} dB_t \tag{11}$$

where the constants κ *denotes respectively the speed of adjustment to the long-term mean* μ *and* α *represents the volatility (see [13]). We will CIR process to estimate the average interest rate for the period during which an LP will be providing liquidity to any one of the protocols discussed in previous sections.*

6.2 *The Variance Gamma Process*

The Variance Gamma is a stochastic process that can be defined considering a standard Brownian motion $B_t, t \geq 0$, and an independent Gamma stochastic process $\Gamma(t; 1, \lambda) = G^{\lambda}(t)$ as follow:

$$X_t^{VG}(\sigma, \lambda, \theta) = \theta G^{\lambda}(t) + \sigma B_{G^{\lambda}(t)}$$

Which is to say this process is essentially a Brownian motion with a drift where the time parameter is replaced by a gamma subordinator.We define a Variance Gamma process in Appendix A.

Consequently, the characteristic function of the Variance Gamma process is given by

$$\phi_{X^{VG}}(u, t) = \left(1 - iu\lambda\theta + u^2\sigma^2\frac{\lambda}{2}\right)^{-\frac{t}{\lambda}}$$

At its core, the VG process can be defined using only three parameters, namely the Brownian motion volatility, σ, the gamma subordinator variance rate, λ, and the Brownian motion drift, θ. Alternatively, in light of the paper by Carr et al. (see [9]), we can define the Variance Gamma process as the difference between two independent Gamma processes, e.g.,

$$X_t^{VG}(\sigma, \lambda, \theta) = \Gamma_p(t; \mu_p, \tau_p) - \Gamma_q(t; \mu_q, \tau_q) \tag{12}$$

where $\Gamma_p(t; \mu_p, \tau_p)$ and $\Gamma_q(t; \mu_q, \tau_q)$ represents two independent Gamma processes, and the parameters satisfying in the Appendix C at the equation (18). The Lévy density for the Variance Gamma process can easily be derived from the Lévy-Khintchine's Theorem as

$$\nu_{VG}(x) = \frac{Ce^{Gx}}{|x|}\mathbf{1}_{\mathbb{R}^-}(x) + \frac{Ce^{-Mx}}{x}\mathbf{1}_{\mathbb{R}^+}(x) \tag{13}$$

the relationship between the C, G, M and $\mu_p, \tau_p, \mu_q, \tau_q$ are presented in the Appendix C at the equation (19).

6.3 Lévy Market based on Variance Gamma process

The Lévy market is a financial model that uses Lévy processes to depict unpredictable jumps and extreme events that occur in real-world financial markets which correspond to Variance Gamma in our discussion. It can handle non-normal distributions, making it useful in volatile markets for asset pricing and risk assessment purposes (see [13]). Lévy market modelling broadens our understanding of complex financial phenomena, allowing for more accurate risk management and derivative pricing. To ensure that the token price process is a martingale, it comes naturally that under the martingale measure obtained by using Girsanov's Theorem, the process that describes the price of tokens, $S_t, t \geq 0$, can be defined as follows:

$$S_t = S_0 e^{m_{new} t + X_t(\sigma,\lambda,\theta)}, \quad \forall t \geq 0 \tag{14}$$

where the term $m_{new} = r + \frac{1}{\lambda}\log(1 - \theta\lambda - \frac{\lambda\sigma^2}{2})$ ensures that the discounted price process is a martingale under the martingale measure.

7 Methodology

The procedure begins with the collection of daily historical price data for each token considered from a given initial date, $date_{in}$, up to a provided present date, $date_{out}$, using a Python script, as well as the United States of America's 10-year bond historical yields from January 1st, 1972, up until the most recent recorded value. The gold standard was abolished in the United States on August 15, 1971, which motivated our decision to set a beginning date for the historical value of the yield.

We utilize the Maximum Likelihood Estimation Method (MLEM) to accurately estimate the parameters of the log of price that reflect the Variance Gamma process. Once each model instance is calibrated to the token prices, we generate $10,000$ corresponding Variance Gamma process samples using a method that takes the calibrated parameters of the model and the token time series. From these samples, we select the $n \in \{1, 2, \ldots, 10000\}$ samples with the lowest Root Mean Square Error (RMSE) on the historical token time series period. Using the Monte Carlo method, we derive the mean process which represents the most likely realization of the token corresponding to the Variance Gamma process.

Using the equations we developed, we calculate an estimation of the impermanent loss from the $date_{out}$ up to the maturity date by considering all the inputs for the impermanent loss calculation that have already taken place.

7.1 Data Collection

A rigorous technique is used when studying a token's historical price data without omitting outliers. This approach recognizes that outliers, however extreme, frequently contain useful market information. Using the Coingecko API, we collect token daily historical prices.

7.1.1 *Primary and Secondary Data*

We have already discussed primary data. Secondary data is also important as it helps us evaluate the performance and resilience of our model in capturing the main patterns present in the token's historical price. Therefore, we consider the period from $date_{out}$ until one day before the LP provides liquidity to one of the mentioned DeFi protocols to be the secondary data period.

8 Model Simulation

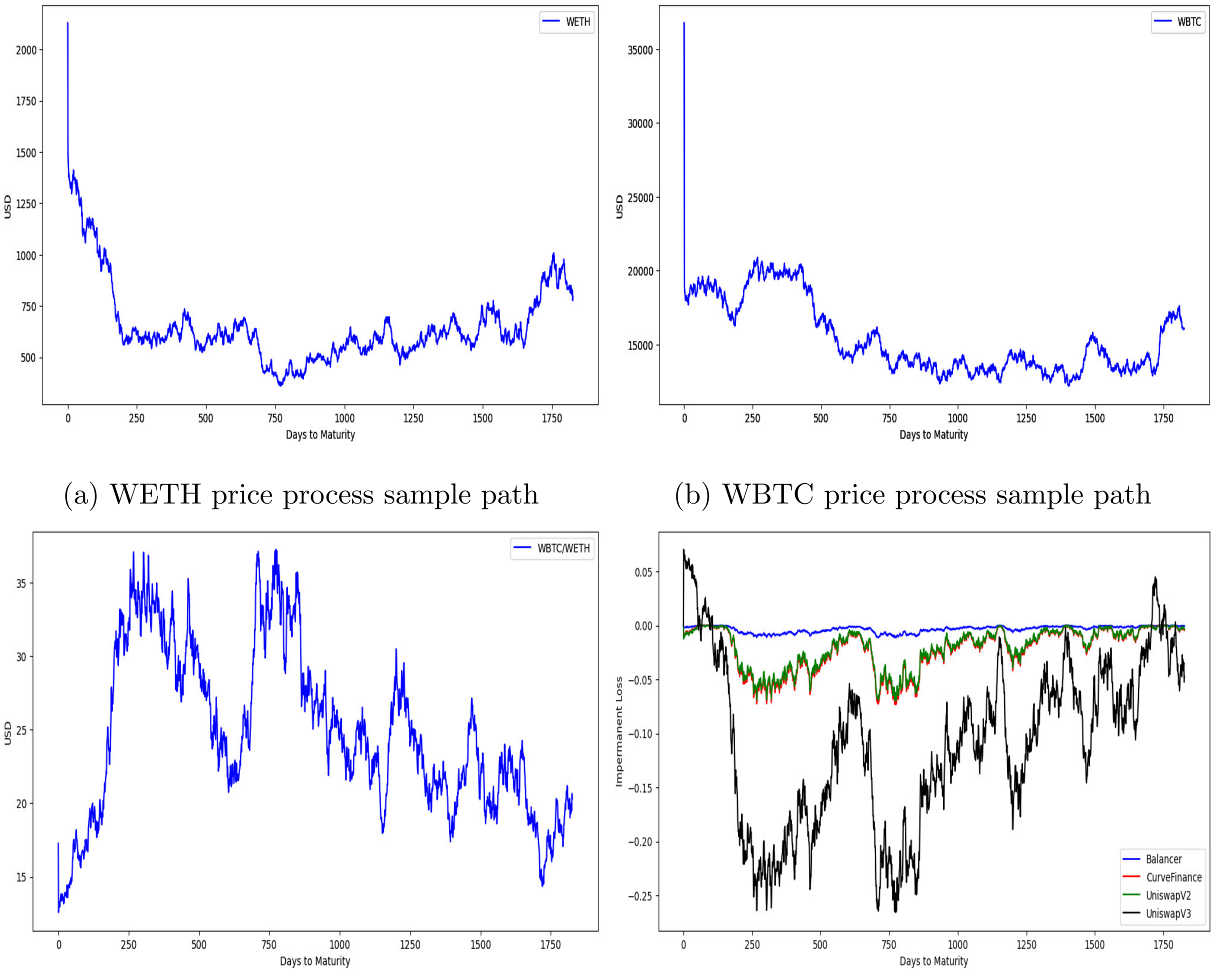
Each stochastic process involved in the building of the Variance Gamma Lévy market model is thoroughly documented in the appendices, along with a method for simulating them.

9 Comparative analysis of Uniswap, Balancer, and Curve Finance

To assess the risk of impermanent loss, we developed a model and considered a set of crypto assets, $WETH/WBTC$. However, it's important to note that not all four DeFi protocols we discuss have these assets. This set is meant to be an illustrative example. Its purpose is to generate multiple possibilities for the token price trajectories. The discussion focuses on the impact of impermanent loss in those market conditions and highlights the Uniswap, Balancer, and Curve Finance protocols. As such this approach is not limited to any specific pairs of tokens.

Let's start our discussion by looking at a time mesh where each point represents a day between the start date, 10/13/2023, and the maturity date, 10/12/2028, which is exactly five years. We have chosen this long maturity date because we want to analyze how long maturities impact impermanent loss.

To create realistic future price trends for each token, we start by setting up two variance gamma Lévy markets, one for each token. Next, we calibrate the parameters of each model to the corresponding token's historical log price data from January 1st, 2017 to October 12, 2023. To analyze various market conditions, we manipulate each parameter of the price process. This forces the price process to reflect typical market conditions, which we then analyze. We generated the price process trajectories for five years in this scenario, resulting in the following outcomes:



(a) WETH price process sample path

(b) WBTC price process sample path

(c) Correspondent relative price

(d) Impermanent Loss Over Time

Figure 2: A realizations of our standard market condition.

The calibrated parameters for WETH and WBTC are $(\sigma; \lambda; \theta) = (1,32833; 0,97886; 0)$, $(\sigma; \lambda; \theta) = (1,01657; 1,08597; 0)$ respectively.

In the Figures 2a, 2b we have realized of price process for WETH and WBTC as the forecast of our model considering the patterns captured from the historical data of each token respectively.

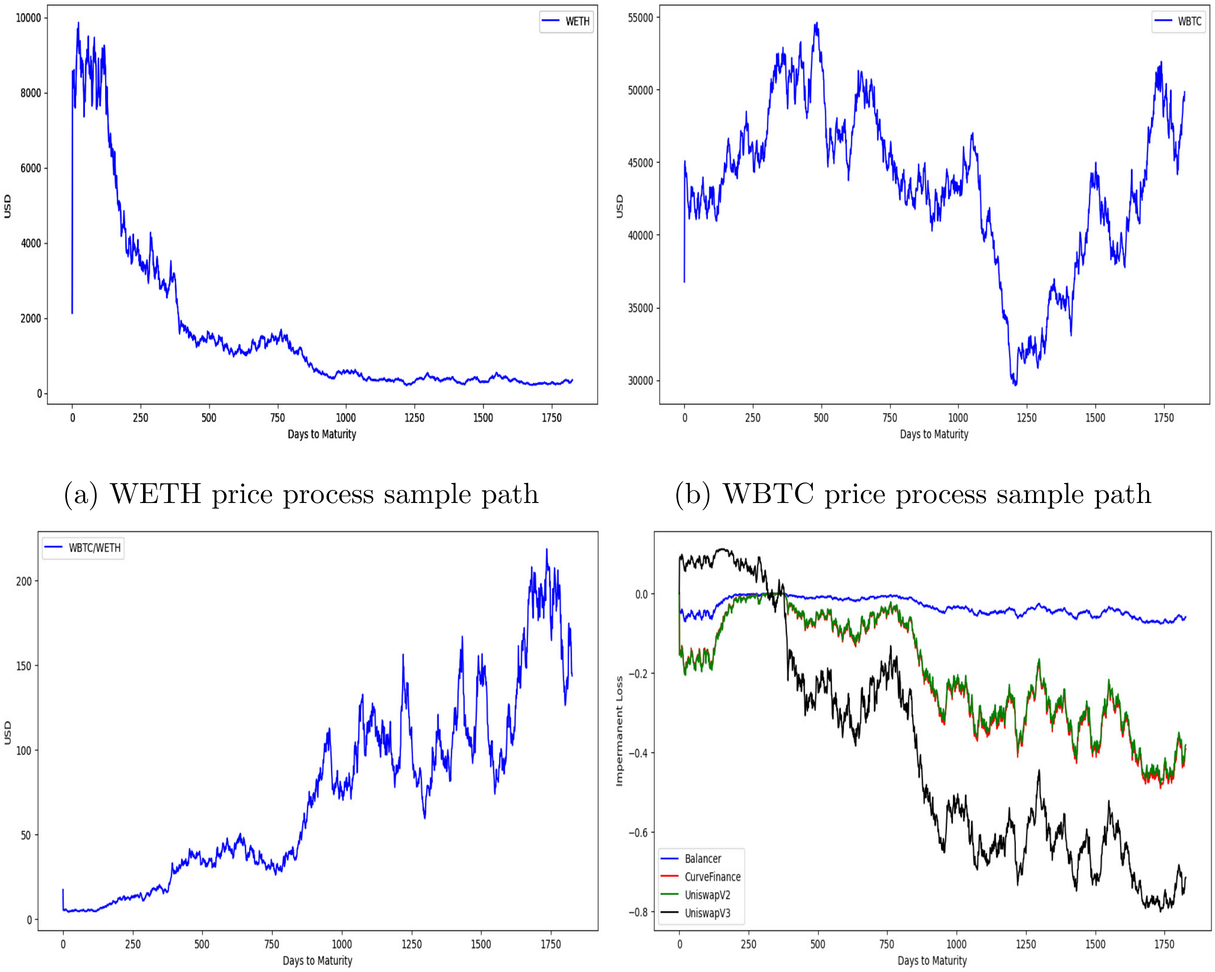
Let us clarify that the reason why the IL trajectories over time for the Curve Finance protocol look similar to that of Uniswap V2 is because we have assumed that both D_t and K_t, defined in the respective equations, are constant. We had to make this assumption since we couldn't access the necessary coefficients on-chain to calculate D_t and K_t as a function of time. It is essential to emphasize that this is the reason why the curves behave in the way they do.

When evaluating the WETH token, it is crucial to notice that the short-term volatility (represented by σ) is roughly 1.5 times larger than the jump intensity parameter (expressed by λ), although not excessively so.

As seen in Figure 2a, this results in more price oscillations with fewer jumps in the short to medium term. However, as the number of days to maturity increases, we will see more volatile fluctuations in prices and frequent jumps. In the case of WBTC Figure 2b, the estimated values for both short-term volatility and jump intensity are almost identical, resulting in a token price with frequent changes in both directions since the combined impacts of these two elements influence the model volatility.

In the short to medium term, Figure 2c demonstrates that there is a significant movement in both directions, but as time progresses, the relative price action becomes less aggressive after the third year. As a result, all projects in the short to medium term are severely affected, except for the Balancer protocols, which provide consistent and optimized IL protection in the liquidity pool settings we are considering. Figure 2d shows that as we approach the maturity time horizon, the IL trajectories of all protocols converge to smaller and smaller values of IL. In this market condition, Uniswap V3 pools are the most affected of them all, with the IL reaching as high as 27% in the first year. This is somewhat reasonable to expect in the short to medium term given the current market condition, with relatively regular and moderate economic shocks.

By increasing the short-term volatility value tenfold in the Figure 3a, and Figure 3b, we get a bigger σ in the model, indicating that asset values are expected to be more volatile in the near future. Short-term volatility can be attributed to factors such as market uncertainty, news events, or speculative trading activity. This indicates that short-term price movements will become more significant and erratic. As sigma increases, liquidity providers will demand greater compensation to cover potential losses or unexpected price swings.



(a) WETH price process sample path

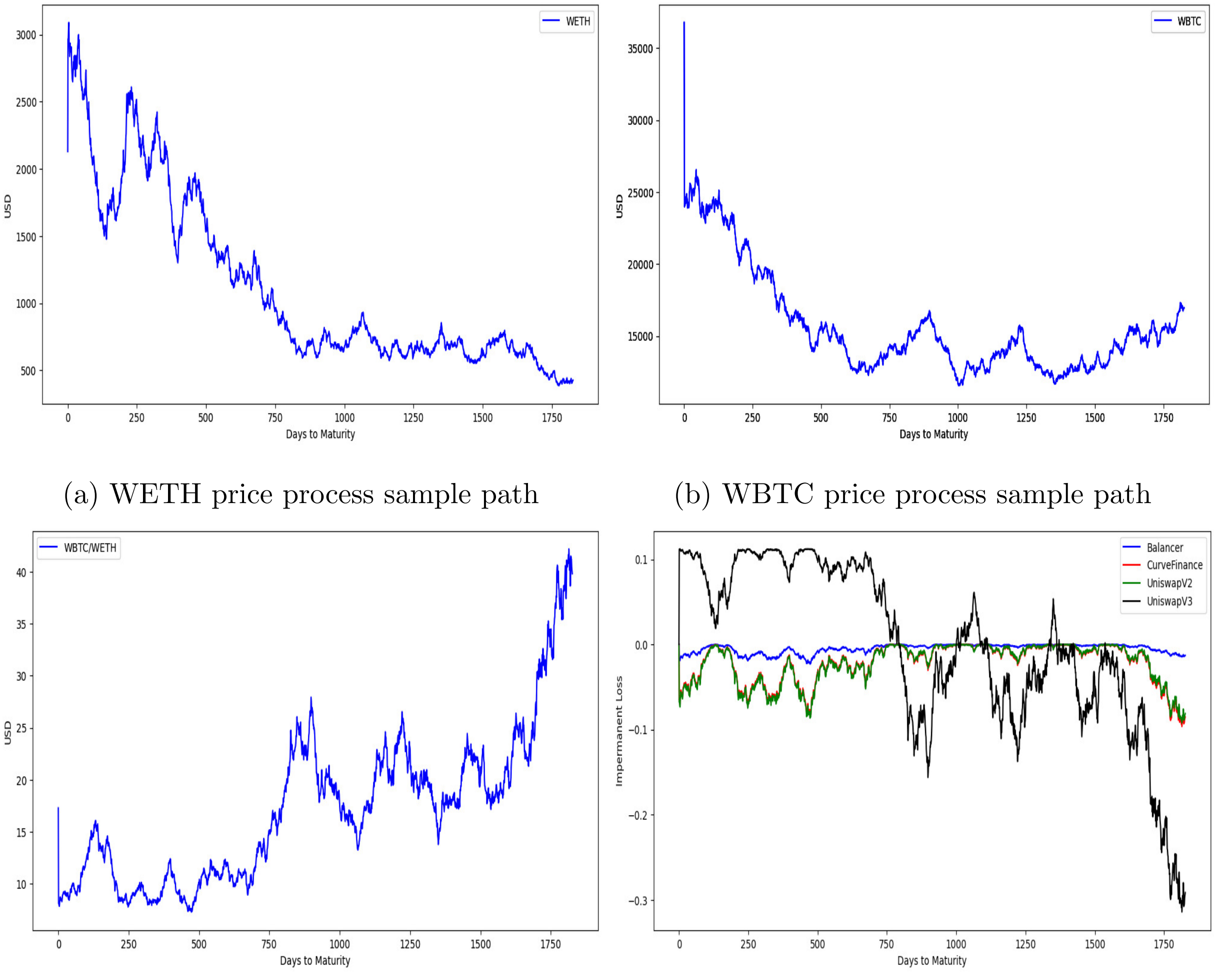
(b) WBTC price process sample path

(c) Correspondent relative price

(d) Impermanent Loss Over Time

Figure 3: The market with higher short-term volatility, $10x\sigma$.

When we examine the IL over time in Figure 3d, we can see the reverse of IL known as the Impermanent gain evidenced by the Uniswap V3 in the short-term. Through this market, an LP would be better suited to supplying liquidity through the Balancer protocol, which still provides more solid IL protection to the LPs, with the Uniswap V3 pools being the most affected. According to 3d as the time to maturity diminishes we can observe the propagated effect of short-term volatility penalizing LPs that preferred either Uniswap V2, V3, or the Curve Finance pools over that of the Balancer.



(a) WETH price process sample path

(b) WBTC price process sample path

(c) Correspondent relative price

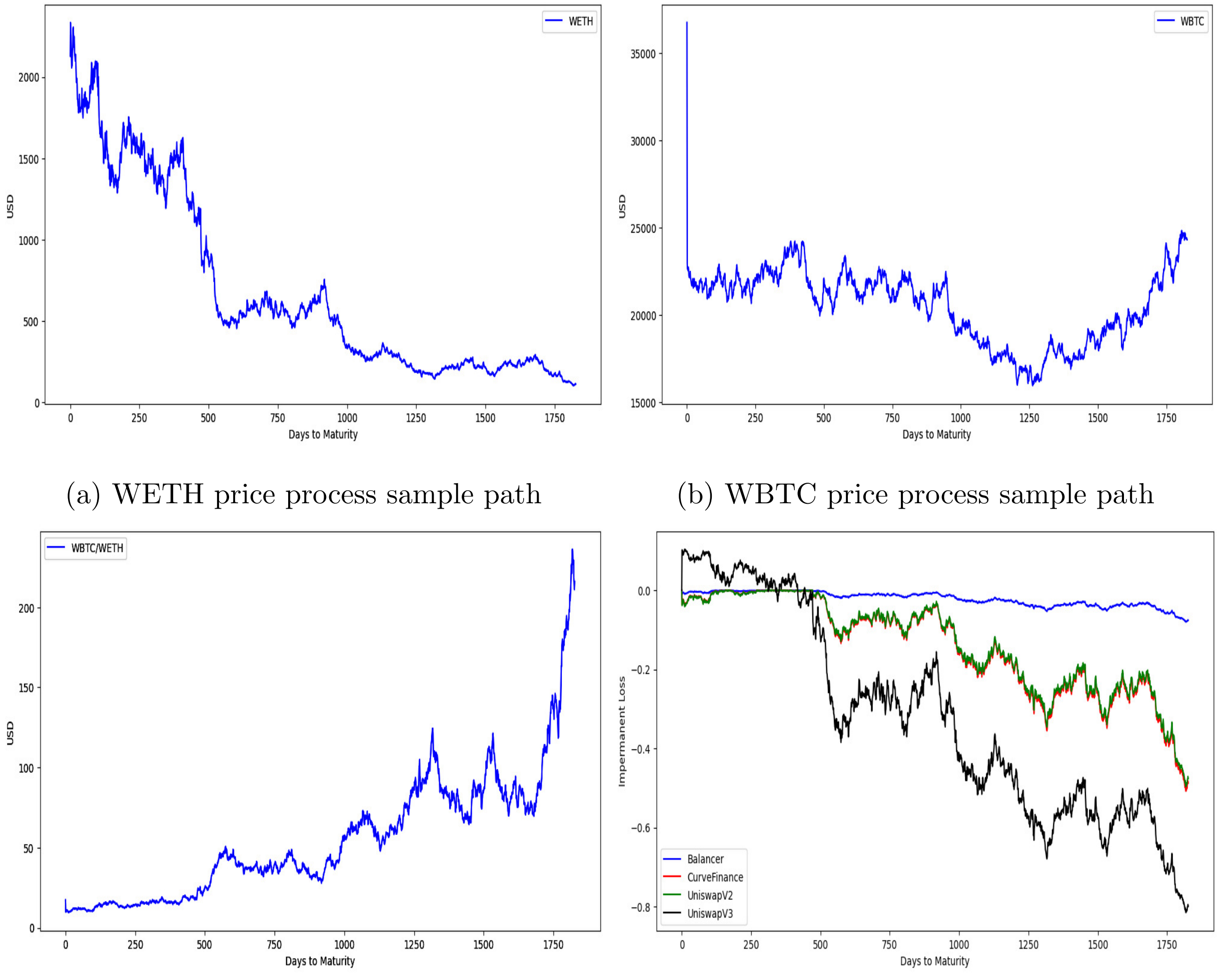
(d) Impermanent Loss Over Time

Figure 4: The market condition with lower short-term volatility, $0,001x\sigma$.

Conversely, if we deflate the short-term volatility a thousandfold because the combined effect of σ and λ controls the model volatility instead of having a consistent low volatile price process, the jump component becomes dominant, as such the sudden price shift is more frequent up to the maturity.

Similarly to the prior market scenario shown in Figure 3, the same conclusion concerning IL can be drawn, except that between two and three and a half years, the Uniswap V3 protocol provides superior IL protection, followed by the Balancer in Figure 4d which assumes the lead thereafter.

Let's multiply the value of the jump component by 10 to originate the market condition



(a) WETH price process sample path

(b) WBTC price process sample path

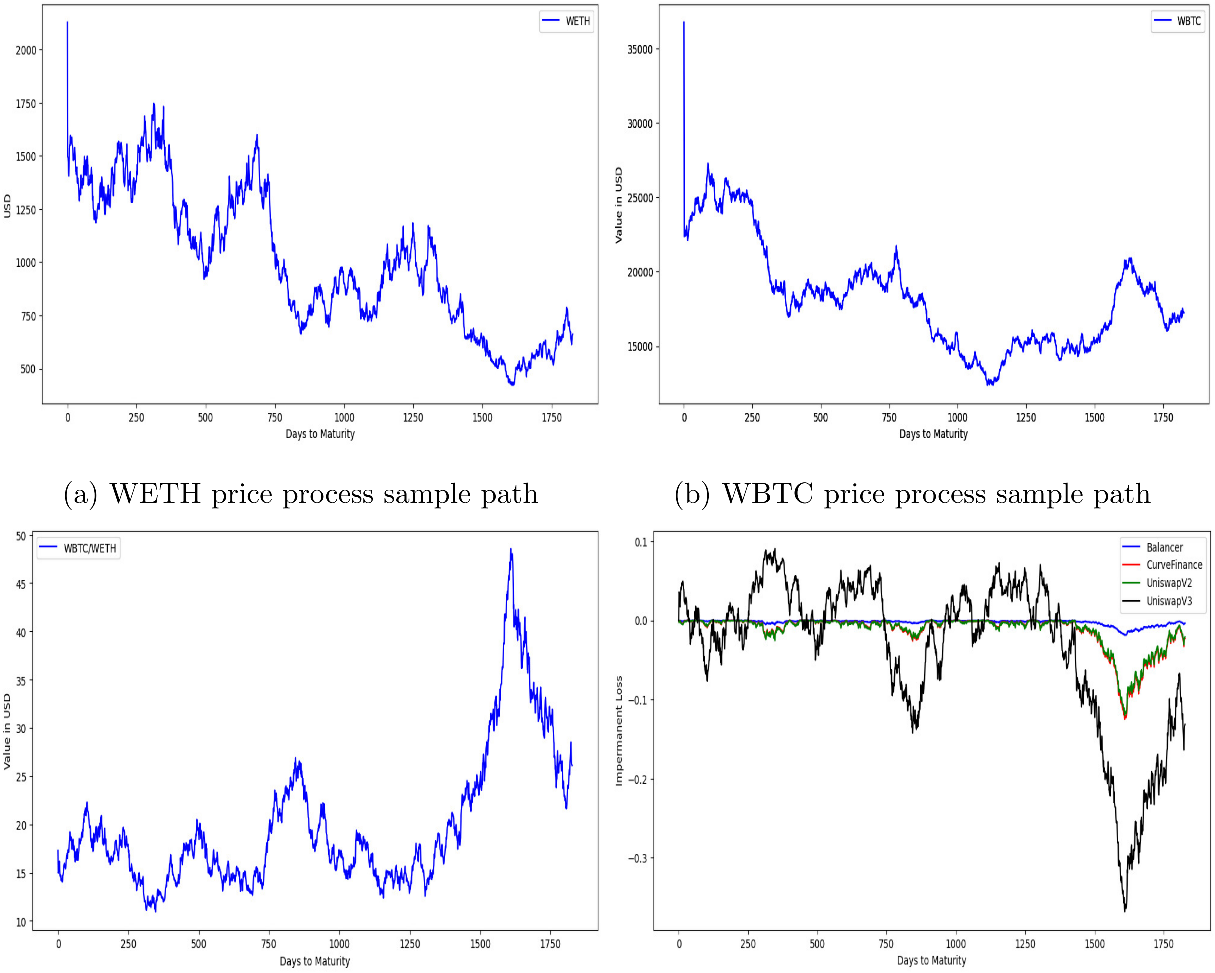
(c) Correspondent relative price

(d) Impermanent Loss Over Time

Figure 5: The market condition with higher jump intensity, $10x\lambda$.

express in the Figure 5. This will result in market conditions marked by huge price spikes, as is typical in the crypto market. Other significant characteristics of this state include speculative trading, poor liquidity in some assets, and a 24-hour trading environment. Market sentiment, news, and trading volumes all contribute to the high value of the jump component. It is important to note that statements or rumours regarding regulatory or market changes might cause uncertainty, which can lead to significant price volatility.

When excessive volatility is produced by a cascade of automated selling orders, flash crashes can occur. These events can result in a series of sharp price drops followed by recoveries, adding to the severity of the jump component in the Lévy market model. Figure 5d shows that these market conditions are highly susceptible to high IL values in the longer timeframe, with the Balancer pool being the least affected.



(a) WETH price process sample path

(b) WBTC price process sample path

(c) Correspondent relative price

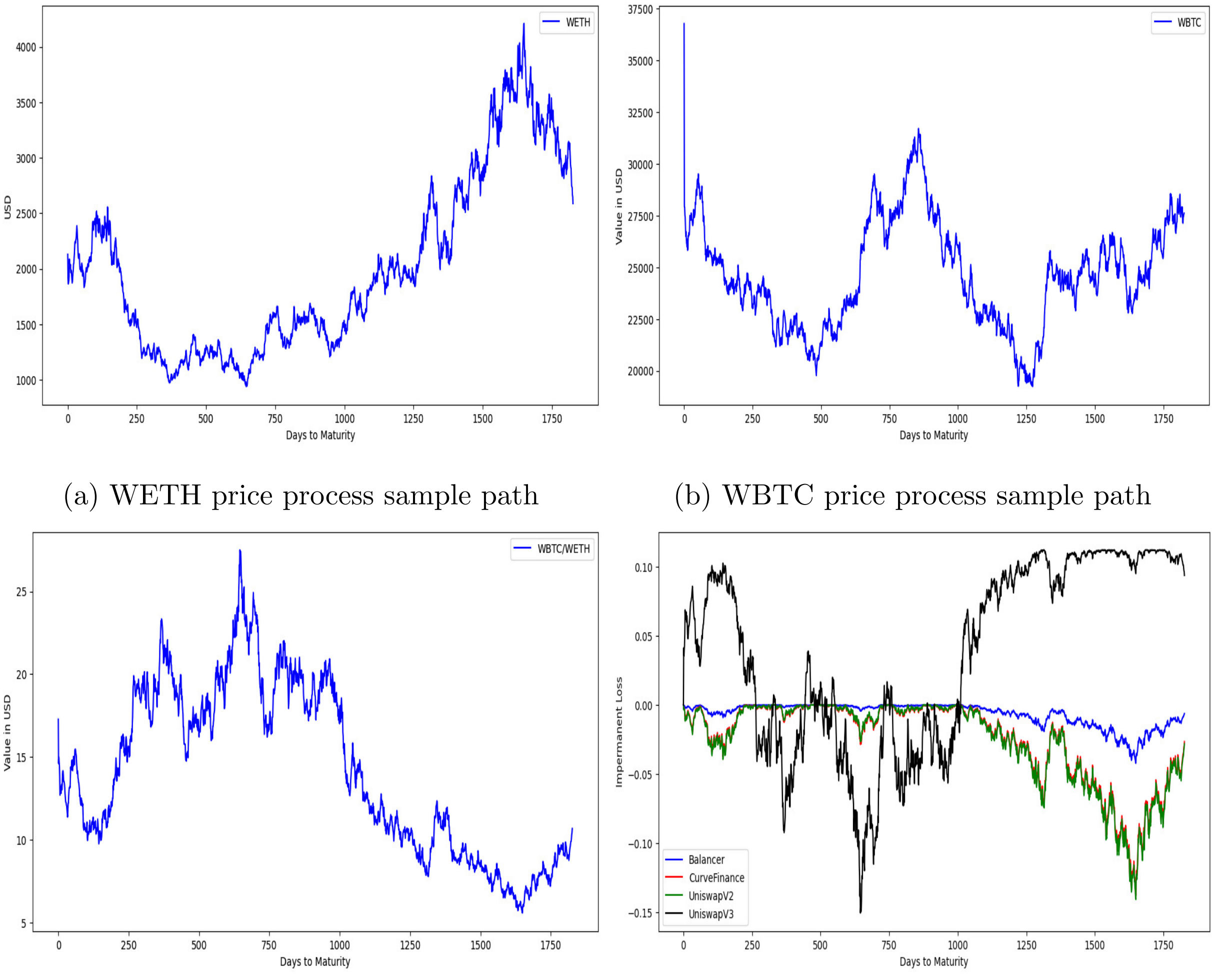
(d) Impermanent Loss Over Time

Figure 6: The market condition with lower jump intensity, $0,001x\lambda$.

When the jump intensity parameter is exceedingly low (as seen in Figure 6), the market experiences few and moderate price fluctuations but large jumps are rare which is visible in Figures 6a, and 6b despite the impact of short-term volatility being more prominent.

In the current market conditions, Uniswap V3 liquidity pools offer better protection against impermanent loss (IL) compared to Balancer pools in the short term, up to approximately four years. However, Balancer pools are a more stable and suitable option for longer time horizons. On the other hand, LPs can expect to face more IL when using Curve Finance and Uniswap V2 pools since these pools are more susceptible to market volatility, as shown in Figure 6d.

Looking at the figure 7d, we can deduce that in our Lévy market model, when the volatility is high, Balancer pools tend to perform better. However, this is only true between the end of the first year and roughly the beginning of the third year, when we consider the combined effect of short-term volatility and the jump component. On the other hand, in the short term and in the longer time horizon, Uniswap V3 outperforms other protocols by providing a better IL edging mechanism to the LPs.



(a) WETH price process sample path

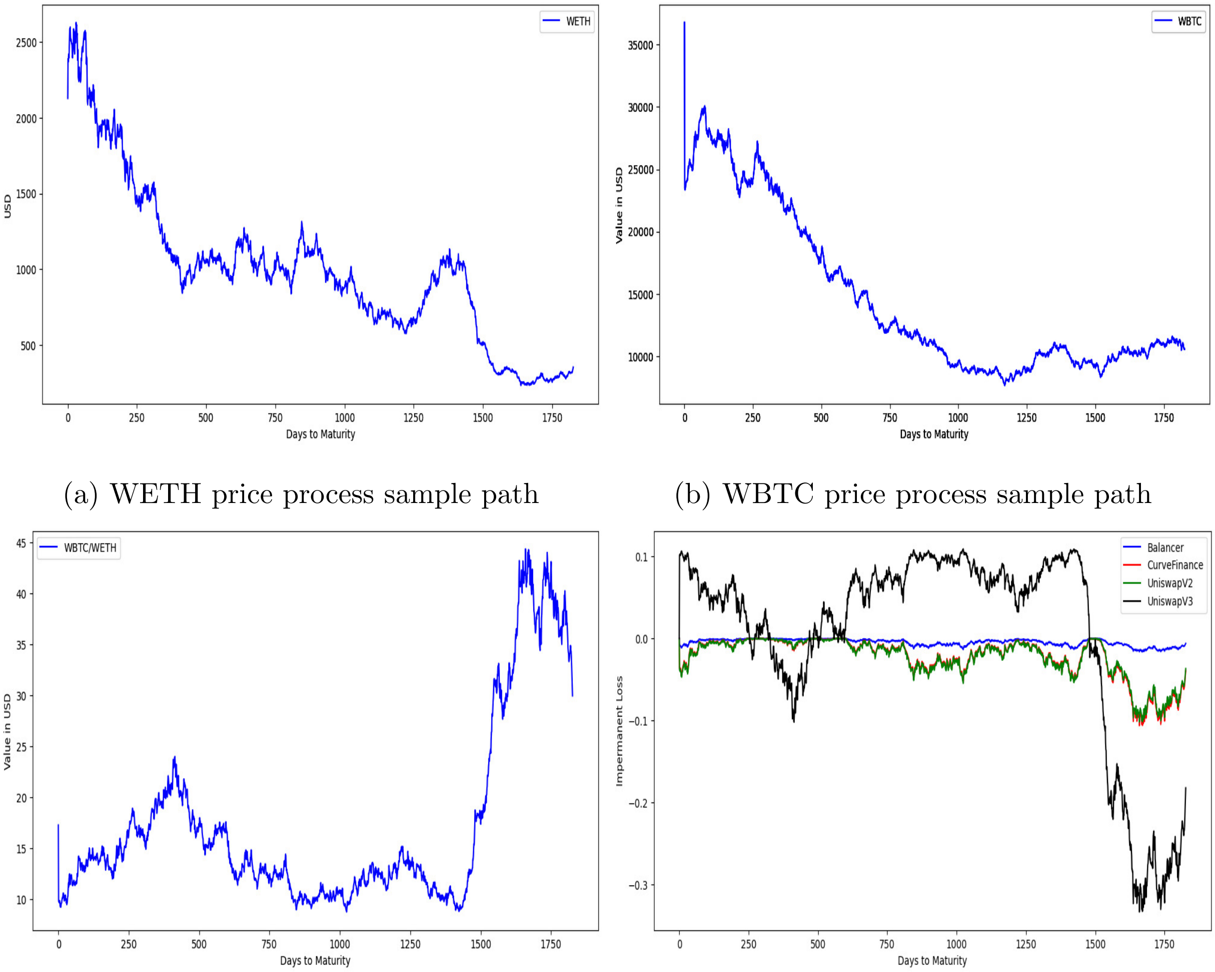
(b) WBTC price process sample path

(c) Correspondent relative price

(d) Impermanent Loss Over Time

Figure 7: High volatility market condition, $10x\lambda, 10x\sigma$.

Conversely, in periods of low volatility in the market, the results presented in Figure 8d suggest that in the short-to-mid time horizon, the Uniswap V3 liquidity pools provide better IL protection, although the Balancer pools are the optimized choice for longer maturity.



(a) WETH price process sample path

(b) WBTC price process sample path

(c) Correspondent relative price

(d) Impermanent Loss Over Time

Figure 8: Low volatility market condition, $0,001x\lambda; 0,001x\sigma$.

Final Considerations

Based on statistical results provided by our Lévy market model with the centred Variance Gamma stochastic process, and after analyzing the impermanent loss risk in the Uniswap V2, Uniswap V3, Balancer, and Curve Finance liquidity pools, it can be concluded that Balancer's 5/95 weighted pools are more likely to be less affected by impermanent loss in general and across the most typical crypto market conditions.

In low-volatility financial markets, the Uniswap V3 protocol stands out as an exception, offering superior protection against IL in the short to medium term. Furthermore, empirical research suggests that even in highly volatile market conditions, the Uniswap V3 protocol provides its LPs with relatively better protection against IL over extended time horizons.

In our research, we analyzed the risk of impermanent loss in liquidity pools provided by various DeFi protocols, taking into account the ”typical” market conditions of the crypto industry. The results of our statistical analysis revealed that Balancer’s 5/95 pools are comparatively less susceptible to impermanent loss.

The Balancer protocol’s dynamic asset allocation and the specific 5/95 pool configuration offer liquidity providers a competitive advantage. Balancer’s design allows for greater control over the composition of the liquidity pool, enabling users to customize their exposure to different assets within the pool. This feature aligns well with the central tenets of the Lévy market model, which emphasize adaptability and risk management. Balancer’s approach to liquidity provision, especially in the context of 5/95 pools, appears to provide better protection against impermanent loss. However, liquidity providers should continually monitor and adapt their strategies based on the evolving landscape of the DeFi market.

A Appendix A: Fundamental concepts about Bitcoin network

In this Appendix section, we define the fundamental notion and terminology required to understand how the Bitcoin Blockchain operates.

Transaction TX

The transactions are a crucial aspect of the Bitcoin system. Every other feature of Bitcoin is designed to make sure that transactions can be created, propagated throughout the network, verified, and eventually added to the global ledger of transactions. Transactions are data structures that contain information about the transfer of value between participants in the Bitcoin system.

Transaction Inputs and Outputs

The fundamental building block of a Bitcoin transaction is a transaction output. Transaction outputs are indivisible chunks of Bitcoin currency, recorded on the blockchain, and recognized as valid by the entire network. Bitcoin full nodes track all available and spendable outputs, known as unspent transaction outputs, or UTXO. The collection of all UTXO is known as the UTXO set and currently numbers in the millions of UTXO. The UTXO set grows as new UTXO is created and shrinks when UTXO is consumed.

Transaction outputs

Transaction outputs consist of two parts:

1) An amount of Bitcoin, denominated in satoshis, the smallest Bitcoin unit;

2) A cryptographic puzzle that determines the conditions required to spend the output.

Transaction Inputs

When creating a transaction, a wallet needs to identify which Unspent Transaction Outputs (UTXOs) it will consume and provide proof of ownership through an unlocking script. To do this, the wallet selects UTXOs it controls that have enough value to cover the requested payment. Depending on the payment amount, the wallet may need to use one or more UTXOs. For each UTXO that will be consumed, the wallet creates an input that points to the UTXO and unlocks it with an unlocking script.

The input contains four elements essentially:

1) A TX ID, referencing the transaction that contains the UTXO being spent;
2) An output index identifying which UTXO from that transaction is referenced;
3) A scriptSig, which satisfies the conditions placed on the UTXO, unlocking it for spending;
4) A sequence number-nonce (deprecated).

Transaction Fees

Bitcoin transaction fees are payments made to transactions to reward miners and maintain the network's operation following consensus rules and methodologies. Higher-fee transactions are preferred by miners since they maximise their revenue while also adhering to the network's consensus process. The fee market is affected by supply and demand for block space. Users can set their prices, although low prices may result in delayed confirmations or low priority. Fee estimation algorithms assist in calculating appropriate fees depending on transaction size and market conditions. Consensus rules ensure that transactions in higher-fee blocks are more likely to be authorised by the network.

Transaction Scripts and Script Language

The Bitcoin transaction script language, Script, is a simple and safe Turing-complete execution language. Using a stack-based execution paradigm and reverse-polish notation, scripts may establish a comprehensive range of criteria and calculations during transaction validation. It is purposely designed to be computationally light to work with a variety of hardware configurations.

The script delivers programmable money, which enables more intricate spending scenarios than ordinary payment transfers. The Turing completeness of Script allows for the creation of advanced smart contracts as well as the execution of complicated calculations within the Bitcoin network.

Digital Signatures

In Bitcoin, a digital signature is a mathematical approach used to demonstrate ownership and authority without revealing the private key. It provides irrefutable proof of permission and assures that the transaction cannot be altered once signed.

Bitcoin uses the ECDSA algorithm for digital signatures. Every transaction input is signed individually using ECDSA. To verify a signature, you need the supporting public key, the serialised transaction, and the signature itself. In Bitcoin, SIGHASH flags are used to specify which parts of the transaction are included in the signature. These flags offer transactional flexibility in several situations.

Public and Private keys

Asymmetric cryptography requires both public and private keys. The public key is widely circulated and used for encryption and verification, ensuring communication privacy and authenticity. It encrypts data and checks digital signatures.

A private key is simply a number picked at random, and it is essential to create signatures that are required to spend Bitcoins by proving ownership of funds used in a transaction.

The public key is calculated from the private key using elliptic curve multiplication, which is irreversible.

Bitcoin Addresses

The Bitcoin address is an alphanumeric identifier used to receive and send Bitcoin cryptocurrency derived from the public key through the use of the one-way cryptographic hashing function.

Merkle Trees

The Merkle tree on the Bitcoin blockchain summarizes all of the transactions in the block. A Merkle tree, also known as a binary hash tree, is a data structure used to quickly summarize and verify large amounts of data. The term ”tree” refers to a branching data structure in computer science nevertheless, these trees are often portrayed upside down, with the ”root” at the top and the ”leaves” at the bottom of an illustration.

Gas

Gas is a virtual fuel in Ethereum that is used to execute smart contracts. The EVM uses an accounting system to measure gas use and limit computer resource usage.

Transparency

Transparency refers to the Bitcoin network's open and public character, in which information about transactions, addresses, and blocks is available to anybody. It is a crucial feature of the Bitcoin protocol that ensures the network's credibility and security.

Through the use of Bitcoin Explorer, an individual can assess all the information ever written on the Bitcoin ledger.

Notice that a blockchain explorer is a web application that operates as a Bitcoin search engine in that it allows anyone to search for addresses, transactions, and blocks and see the relationships and flows between them.

Popular blockchain explorers include- Bitcoin Block Explorer, BlockCypher Explorer, blockchain.info etc.

B Appendix B: Stochastic Processes

Definition 3. *A stochastic process is a collection of random variables* $\{X(t, \omega), t \in \Lambda, \omega \in \Omega\}$ *defined in the product space* $\Lambda x \Omega$ *such that:*

i) $\forall t \in \Lambda$, $X(t, \cdot)$ *is a random variable;*

ii) $\forall \omega \in \Omega$, $X(\cdot, \omega)$ *is a* $\mathscr{F}$*-measurable function.*

Where Λ *represents a Borel-measurable set. Usually, when* Λ *is countable set, i.e;* $\Lambda \subseteq \mathbb{N}_0$, *we say that* $X(\cdot, \cdot)$ *is a discrete-time stochastic process. Similarly when* $\Lambda \subseteq \mathbb{R}_0^+$, *we designate the process as a continuous-time stochastic process.*

As an example, consider the following types of stochastic processes:

Definition 4. *A Stochastic process $B(\cdot,\cdot)$ is known as Brownian motion or Wiener process if it satisfies the following proprieties:*

1) for each $\omega \in \Omega$ (fixed), we have:

1.1) $P\left(\{B(0,\omega) \neq 0\}\}\right) = 0$;

1.2) considering $0 \leq s \leq t$, the random variable $B(t,\omega) - B(s,\omega) = B(t-s,\omega)$ is a Gaussian with mean 0 and variance $t-s$;

1.3) for each partition $0 = t_0 \leq t_1 \leq \ldots \leq t_n$, $n \in \mathbb{N}$, the increments $B(t_1,\omega), B(t_2-t_1,\omega), \ldots, B(t_n-t_{n-1},\omega)$ are independents;

1.4) $P\left(\{B(\cdot,\omega)\ \textit{is continuous}\}\right) = 1$

This process is extensively utilized in mathematical finance, particularly in the Black-Scholes model of the financial market, and is also a vital element in constructing a pure Lévy stochastic process.

B.1 Brownian Motion

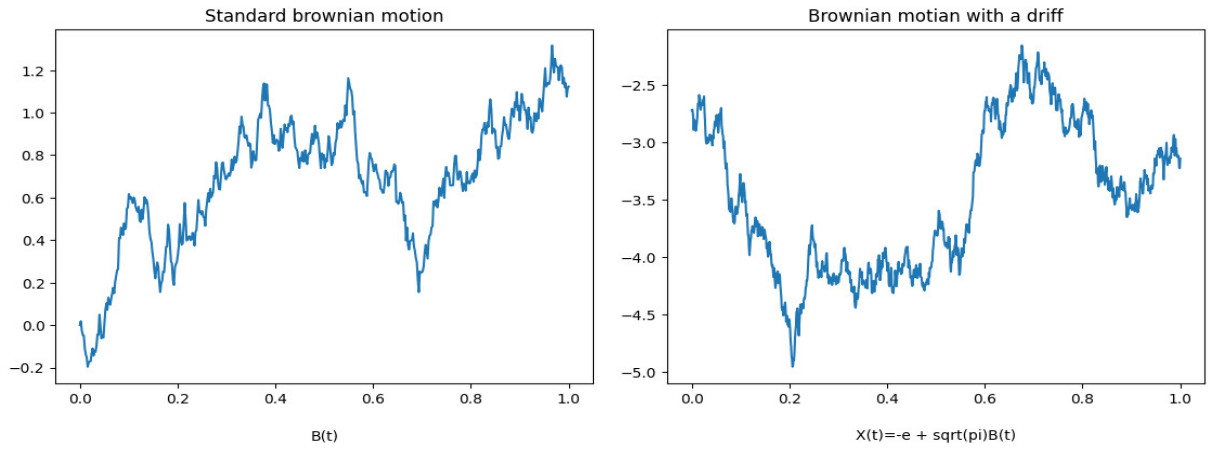
When we defined Brownian motion, we noticed that this stochastic process has independent and stationary increments that are Normally distributed. Furthermore, we know that the Gaussian distribution is fully characterized by its parameters, which were also provided in the definition of Brownian motion. Based on this insight, we can simulate a one-dimensional Wiener process, B_t, by considering a time grid that is defined as:

$$\Delta t_{\mathrm{i}} = t_{\mathrm{i}} - t_{\mathrm{i}-1},\ \text{for each } i = 1, 2 \ldots, n \in \mathbb{N}$$

Moreover, we will take $B_0 = 0$ and provide an approximation of a Wiener process at each point in this time grid as follows:

$$B_{\mathrm{t_i}} := B_{\mathrm{t_{i-1}}} + \sqrt{\Delta t_{\mathrm{i}}} Z_{\mathrm{i}}, \quad Z_{\mathrm{i}} \sim \mathscr{N}(0,1)$$

The figure below provides an example of a sample path generated through the simulation technique described earlier:



(a) Example 1

(b) Example 2

Figure 9: Brownian Motion sample paths

Another example is given as follows:

Definition 5. *Consider* $\omega \in \Omega$ *(fixed),* $n \in \mathbb{N}$ *and* $t_1, \ldots, t_n \in \Lambda \subseteq \mathbb{R}_0$ *such that* $t_1 \leq \ldots \leq t_n \leq t$. *A stochastic process* $X_t = X(t, \omega)$ *is said to be a* Marcov *process if it satisfy to condition:*

$$P\left(X_t \in A | X_{t_1} = x_1, X_{t_2} = x_2, \ldots, X_{t_n} = x_n\right) = P\left(X_t \in A | X_{t_n} = x_n\right)$$

for any Borel-measurable set A, $t_1, \ldots, t_n \in \Lambda$ *and* $x_1, \ldots, x_n \in \mathbb{R}$.

Markov processes form the backbone for the Markov chain Monte Carlo method, a type of stochastic simulation approach used to sample from intricate probability distributions. These methods have widespread applications in various fields such as Bayesian statistics, physics, economics, chemistry, and signal processing.

Definition 6. *A Stochastic process* $N(\cdot, \cdot)$ *defined in the probability space* $(\Omega, \mathscr{F}, P)$, *and with state space* $\Lambda \in \mathbb{N}_0$ *is said to be Poisson process with intensity* $\lambda > 0$ *if the following condition holds:*

1) for each $\omega \in \Omega$ *(fixed), we have:*

1.1) $P\left(\{N(0, \omega) = 0\}\right) = 1$;

1.2) $\forall 0 \leq s < t$, *the random variable* $N(t, \omega) - N(s, \omega)$ *has Poisson distribution with intensity* $\lambda(t - s)$, *more specifically:*

$$P\left(N(t) - N(s) = n\right) = \frac{\left(\lambda(t-s)\right)^n e^{-\lambda(t-s)}}{n!}, \quad \forall n \in \mathbb{N}_0$$

1.3) for each time partition $0 = t_0 \leq t_1 \leq \ldots \leq t_n$, $n \in \mathbb{N}$, *the increments* $N(t_1), N(t_2) - N(t_1), \ldots, N(t_n) - N(t_{n-1})$ *are independents;*

1.4) $P\left(\{N(\cdot,\omega)\text{ is continuous on the right}\}\right) = 1$

The Poisson process has been widely applied to simulate the occurrence of separate and seemingly unrelated events. It is crucial to Queueing theory, a branch of probability theory that deals with building suitable stochastic models to capture the arbitrary arrival and departure of certain occurrences.

Definition 7. *Consider a stochastic process* $\{S(t), t \geq 0\}$ *defined in the probability space* $(\Omega, \mathscr{F}, P)$, *as follow:*

$$S(t) = \sum_{i=1}^{N_t} X_i$$

where

(1) $\{N_t, t \geq 0\}$ *is a Poisson process with intensity* $\lambda > 0$*;*

(2) $\{X_i, i \in \mathbb{N}\}$ *is a sequence of independents and identically distributed (iid) random variables;*

(3) the processes $\{N_t, t \geq 0\}$ *and* $\{X_i, i \in \mathbb{N}\}$ *are independents;*

Then $S(t)$ *is said to be a Compound Poisson process.*

It is important to notice that the process we described as N_t *is a Poisson process with parameter* λt*, i.e.,*

$$P\left(N_t = n\right) = \frac{(\lambda t)^n \, e^{-\lambda t}}{n!}, \quad \forall n \in \mathbb{N}_0$$

To simulate a one-dimensional Poisson process N_t*, we first simulate an independent sequence of Exponential random numbers* $\{e_n, n = 1, 2, \ldots\}$ *with intensity* λ *as:*

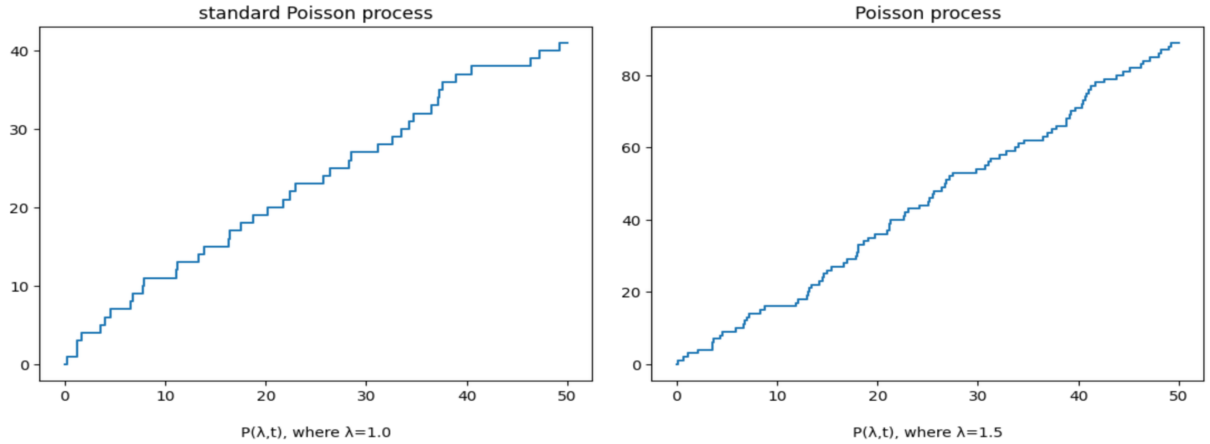
$$e_{\mathrm{n}} = -\frac{\log(u_{\mathrm{n}})}{\lambda}, \quad \forall n \in \mathbb{N}, \quad u_{\mathrm{n}} \sim Unif(0,1)$$

With these inputs, we can now define the sequence of calls arrival times as:

$$s_0 = 0, \text{ and } s_{\mathrm{n}} = s_{\mathrm{n-1}} + e_{\mathrm{n}}, \quad \forall n \in \mathbb{N}$$

The number of phone calls received in a call center in the time interval $[0,t]$*, will be given as:*

$$N_0 = 0, \quad N_{t_{\mathrm{i}}} = \sup_{k\in\mathbb{N}} \left\{s_{\mathrm{k}} \leq t_{\mathrm{i}}\right\}, \text{ for each } t_0 < t_1 < \cdots < t_n$$



(a) Example 1

(b) Example 2

Figure 10: Sample path of Poisson random variable built using the described approach.

Gamma process

Gamma process, $\Gamma(t, \omega; r, \lambda)$ can be defined as the stochastic process with the following proprieties: if it satisfies the following proprieties:

1) for each $\omega \in \Omega$ (fixed) we can use the notation $\Gamma(t, \omega; r, \lambda) = \Gamma(t; r, \lambda)$, and we have:

   1.1) $P\left(\{\Gamma(0; r, \lambda) \neq 0\}\right) = 0$;

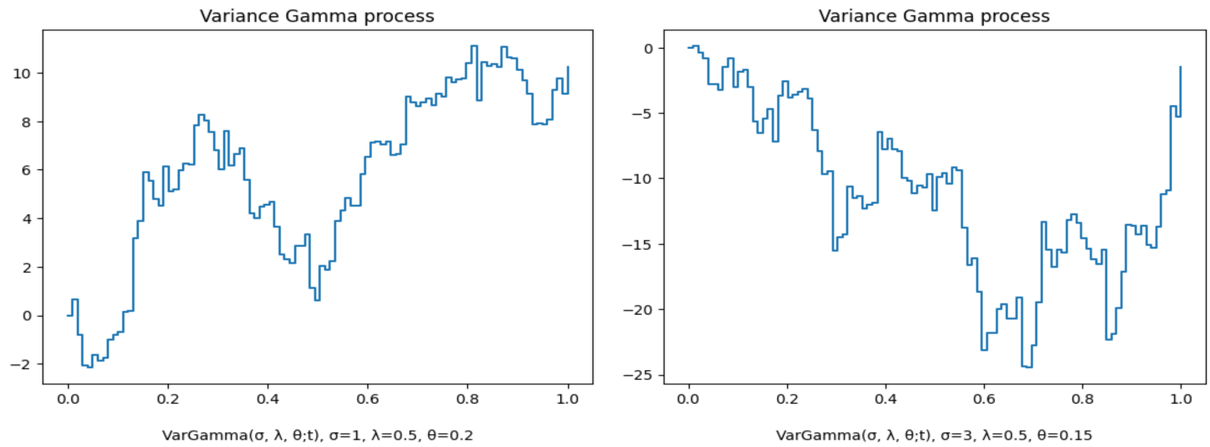
   1.2) considering $t, h > 0$, the random variable $\Gamma(t+h; r, \lambda) - \Gamma(t; r, \lambda)$ is a random variable whose law is absolutely continuous with respect to the Lebesgue measure with density:

$$f(x) = \left(\frac{r}{\lambda}\right)^{\frac{r^2h}{\lambda}} \frac{x^{\frac{r^2h}{\lambda}-1} e^{-\frac{r}{\lambda}x}}{\Gamma(\frac{r^2h}{\lambda})}, \quad x \in \mathbb{R}^+;$$

   1.3) for each partition $0 = t_0 \leq t_1 \leq \ldots \leq t_n$, $n \in \mathbb{N}$, the increments $\Gamma(t_1; r, \lambda), \Gamma(t_2 - t_1; r, \lambda), \ldots, \Gamma(t_n - t_{n-1}; r, \lambda)$ are independents.

It follows that the characteristic function of $\Gamma(t; r, \lambda)$ is given by:

$$\Phi(u; t) = \left(1 - iu\frac{\lambda}{r}\right)^{-\frac{r^2t}{\lambda}}$$



(a) Example 1

(b) Example 2

Figure 11: Two realizations of a Variance Gamma process

Let us proceed to characterize this class of stochastic processes using characteristic functions.

Definition 8. *We define the characteristic function of a random variable X with values in the set $\mathbb{R}$, and distribution μ as function $\Phi_X : \mathbb{R} \mapsto \mathbb{C}$ such that:*

$$\Phi_X(u) := \mathbb{E}\left[e^{iuX}\right] = \int_{\mathbb{R}} e^{iux}\mu(dx), \quad \forall u \in \mathbb{R}$$

It's a well-known fact that there is a one-to-one correspondence between a distribution function and the characteristic function that's linked to the relevant random variable. This means that we can say with confidence that the characteristic function provides a complete characterization of the distribution of the associated random variable.

Definition 9. *We define an* Infinitely Divisible *distribution as a distribution μ such that for each $n \in \mathbb{N}$, there is a distribution μ_n whose characteristic function, Φ_{μ_n}, satisfies:*

$$\Phi_\mu(u) = \left(\Phi_{\mu_n}(u)\right)^n, \quad \forall u \in \mathbb{R}$$

It is important to emphasize that we can derive an implicit solution to the dynamic of the the CIR process defined in equation (11) as it is equivalent to the following:

$$\begin{aligned}
&dy_t + \kappa y_t \mathrm{dt} = \kappa\mu \mathrm{dt} + \alpha\sqrt{y_t} dB_t \Leftrightarrow \\
&e^{\kappa t} dy_t + e^{\kappa t}\kappa y_t \mathrm{dt} = e^{\kappa t}\kappa\mu \mathrm{dt} + \alpha e^{\kappa t}\sqrt{y_t} dB_t \Leftrightarrow \\
&d\left(e^{\kappa t} y_y\right) = e^{\kappa t}\kappa\mu \mathrm{dt} + \alpha e^{\kappa t}\sqrt{y_t} dB_t \Leftrightarrow \\
&y_t = y_0 e^{-\kappa t} + \mu\left(1 - e^{-\kappa t}\right) + \alpha\int_0^t e^{(s-t)\kappa}\sqrt{y_s} dB_s, \quad \forall t \geq 0.
\end{aligned} \tag{15}$$

which is a Gaussian random variable for each $t \geq 0$, with mean and variance, according to Itö's isometry, given by:

$$y_0 e^{-\kappa t} + \mu\left(1 - e^{-\kappa t}\right), \quad \alpha^2 \int_0^t e^{2(s-t)\kappa} y_s ds$$

We can go further and express both the variance and the third non-centered moment of the CIR process as it follows that:

$$\begin{aligned}
Var[y_t] &= \frac{\alpha^2}{\kappa}\left(y_0 e^{-\kappa t}(1 - e^{-\kappa t}) + \frac{\mu}{2}(1 - e^{-\kappa t})^2\right); \\
\mathbb{E}[y_t^3] &= R_t^3 + \frac{3\alpha^2}{4\kappa^2} R_t \left(1 - e^{-2\kappa t}\right)^2.
\end{aligned}$$

Where for each $t \geq 0$, the function R_t is defined as $R_t = y_0 e^{-\kappa t} + \mu\,(1 - e^{-\kappa t})$.

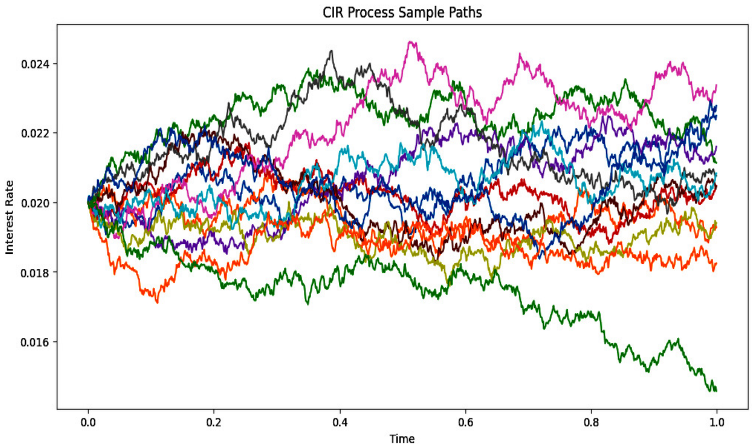
CIR process simulation

The simulation of the CIR stochastic process, we can use the Euler scheme at an equally spaced time points $\{n\Delta\mathrm{t}, n = 1, 2, \ldots\}$ as follow:

$$y_{\mathrm{n}\Delta\mathrm{t}} = y_{(\mathrm{n}-1)\Delta\mathrm{t}} + \kappa\left(\mu - y_{(\mathrm{n}-1)\Delta\mathrm{t}}\right)\Delta\mathrm{t} + \alpha\sqrt{\Delta\mathrm{t}y_{(\mathrm{n}-1)\Delta\mathrm{t}}} Z_{\mathrm{n}}, \quad Z_{\mathrm{n}} \sim \mathscr{N}(0,1)$$

CIR Parameters

Let us consider the following parameters acquired by the Least Squares Estimation method on 10-year T-bond interest rate historical values since January 1 of the year 1972 up until July 1^{st} of the year 2023 as a reference, therefore in this setting, the CIR process trajectories are presented in the next figure



C Appendix C: Key Terms and Definitions

Consider the following definition:

Definition 10. *We can define The main Kummer confluent hypergeometric function as*

$$ {}_1F_1(a,b;z) = \sum_{n\in\mathbb{N}_0} \frac{(a)_n}{(b)_n}\frac{z^n}{n!} = \frac{\Gamma(b)}{\Gamma(a)} \sum_{n\in\mathbb{N}_0} \frac{\Gamma(a+n)}{\Gamma(b+n)}\frac{z^n}{n!} \tag{16} $$

where $(a)_n = \prod_{i=0}^{n-1}(a+i) = \dfrac{\Gamma(a+i)}{\Gamma(a)}$ *is usually known as Pochhammer symbol and by convention,* $(a)_0 = 1$*. It fallows that for each* $h \in \mathbb{N}$

$$ {}_1F_1(a+h,r;z) = e^z \sum_{i=0}^{h} \binom{h}{i} z^i \frac{\Gamma(r)}{\Gamma(r+i)} $$

Definition 11. *We say a random variable, X, follows a non-centered Gamma distribution if its law is a measure which is absolutely continuous with respect to the Lebesgue measure with density given by:*

$$ f_X(x) = \frac{\lambda^r}{\Gamma(r)} e^{-\lambda x} x^{r-1} e^{-\delta\lambda} {}_1F_1(r,\lambda;\delta\lambda^2 x) \tag{17} $$

where δ *denotes the non-centrality parameter. In these circumstances the usual Gamma random variable comes as a particular case when* $\delta = 0$*, and we denote this case as* $X \sim Gamma(r,\lambda)$ *with* $r,\lambda \in \mathbb{R}^+$

We are considering the definition of the gamma function as follows:

$$ \Gamma(r) = \int_{\mathbb{R}^+} e^{-x} x^{r-1} dx, \quad r \in \mathbb{R}^+ $$

Notice that the representation provided in equation 12 is only well defined if, and only if all the following conditions are satisfied:

$$\frac{\mu_p^2}{\tau_p} = \frac{\mu_q^2}{\tau_q} = \frac{1}{\lambda}; \ \frac{\tau_p \tau_q}{\mu_p \mu_q} = \frac{\sigma^2 \lambda}{2}; \ \frac{\tau_p}{\mu_p} - \frac{\tau_q}{\mu_q} = \theta\lambda. \tag{18}$$

and it is important to emphasize that

$$C = \frac{1}{\lambda}, G = \left(\sqrt{\frac{\theta^2\lambda^2}{4} + \frac{\sigma^2\lambda}{2}} - \frac{\theta\lambda}{2}\right)^{-1}, M = \left(\sqrt{\frac{\theta^2\lambda^2}{4} + \frac{\sigma^2\lambda}{2}} + \frac{\theta\lambda}{2}\right)^{-1} \tag{19}$$

(see [8])

D Appendix D: Parameters Analysis

D.1 Parameters effects on the Variance Gamma process

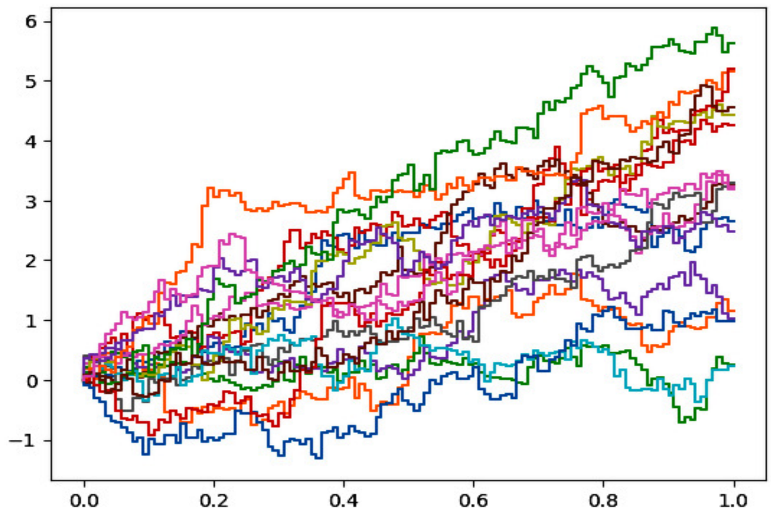
In this section, we are going to examine the fundamental influence each of the three parameters, σ, λ, θ, has in the correspondent Variance Gamma stochastic processes to further confirm the process theoretical expected behaviour and properties. We will proceed by amplifying and decreasing each parameter by a factor of 5 (or 10 in case of σ and 15 in case of θ) and present the correspondent 17 sample path of the correspondent Variance Gamma process.

For sake of clarity, in the following figure, we generate 17 trajectories from the Variance Gamma process on a mesh made of 121 points, $X_t^{VG}(0,18;0,7367;0,04056),\ 0 \leq t \leq 1$, as our starting point with parameters from Schoutens pg 82 (see [13]).

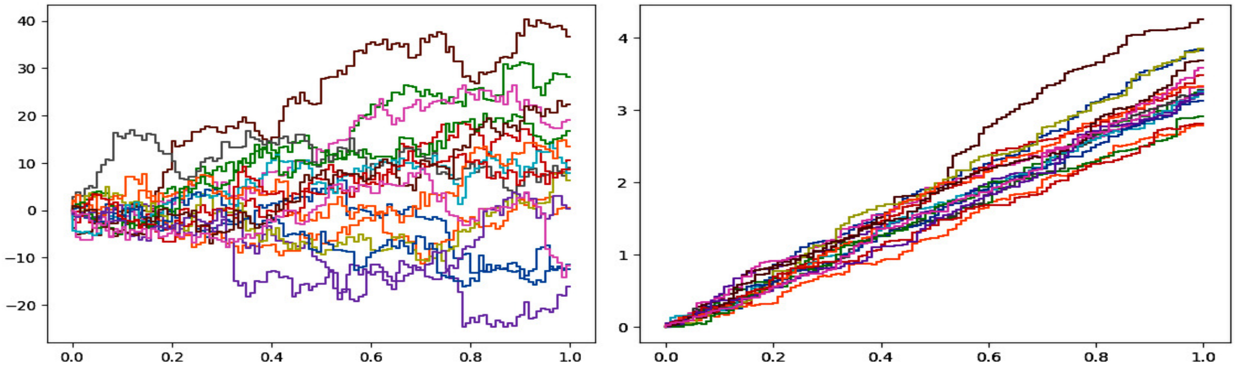
D.1.1 ***Volatility Parameter***

Effect in VG Process: As the VG process does not implicitly have a diffusion component such as geometric Brownian motion (GBM). Through process jumps, the VG process accounts for volatility.

In VG modelling, the volatility impact is handled indirectly via the jump component in the Lévy measure and the variance parameter σ. As these two parameters increase they lead to higher jumps, increasing the token's price volatility.



(a) Variance Gamma trajectories

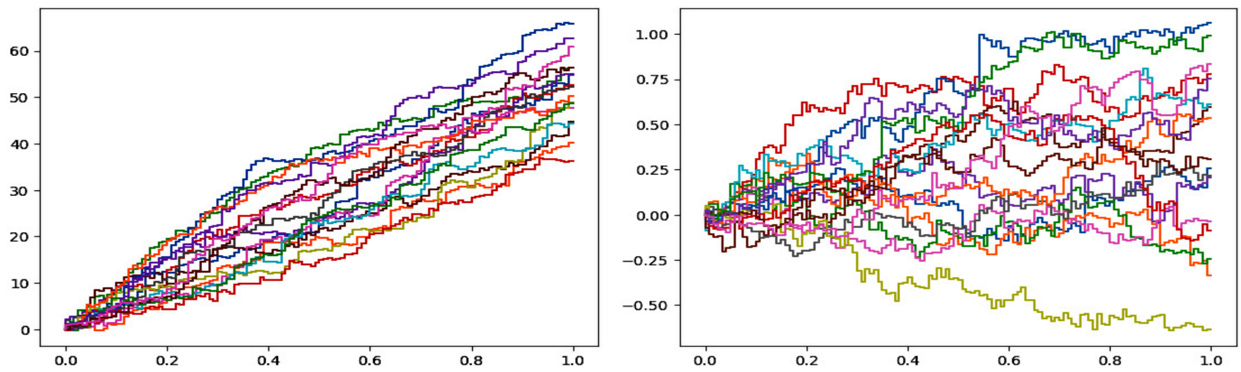


(a) The effect of increasing σ

(b) The effect of decreasing σ

D.1.2 *Jump Parameter*

Effect in VG Process: The jump parameter λ reflects the VG process's jump component. It regulates the intensity of the jumps. A large value of λ suggests more frequent and greater leaps. As the value of λ increases, the price process becomes more volatile and results in larger and more frequent price spikes. This is why VG models are well-suited for capturing significant price movements in financial markets. For instance, VG models are particularly useful for predicting stock price spikes, which is one of the most well-known characteristics of the crypto market.

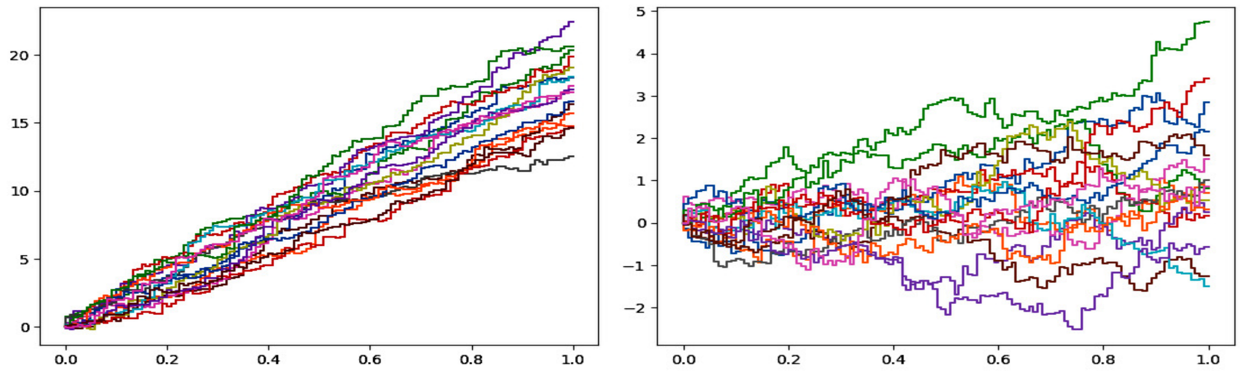


(a) The effect of increasing λ

(b) The effect of decreasing λ

D.1.3 *Drift Parameter*

Effect in VG Process: the drift parameter θ indicates the VG process's average rate of return or linear growth component. A positive drift indicates an upward tendency, whereas a negative drift leads to a downward trend. The following figure emphasizes the control this parameter has on the correspondent VG process:



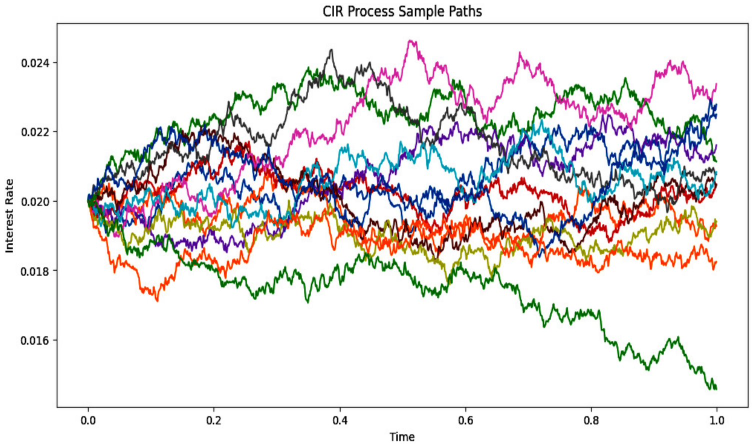
(a) The effect of increasing θ

(b) The effect of decreasing θ

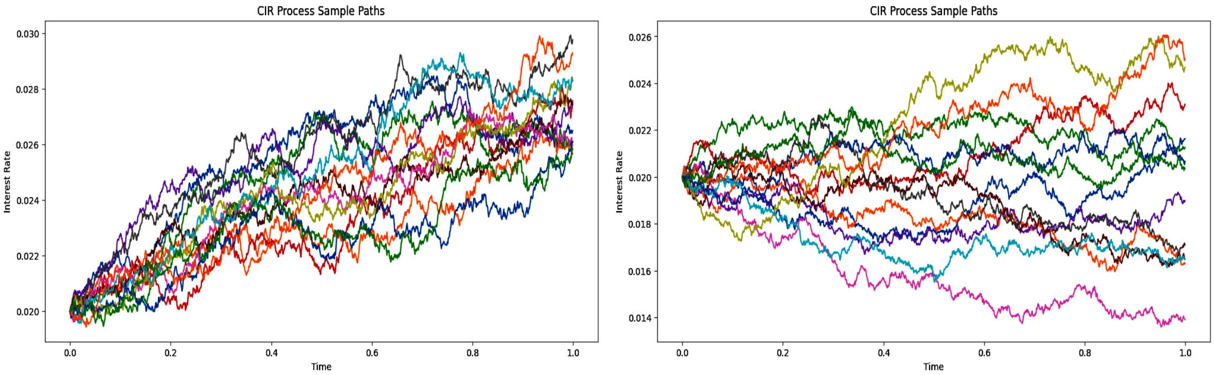
In the context of financial modelling, positive drift indicates that the asset's price is expected to grow over time on average, making it a great choice for modelling assets with historically positive returns. A negative drift, on the other hand, shows a decrease in the price of the asset over time.

D.2 Parameters effects on the CIR process

Let us consider the following parameters acquired by the Least Squares Estimation method on 10-year T-bond interest rate historical values since January 1 of the year 1972 up until July 1^{st} of the year 2023 as a reference, therefore in this setting, the CIR process trajectories are presented in the next figure



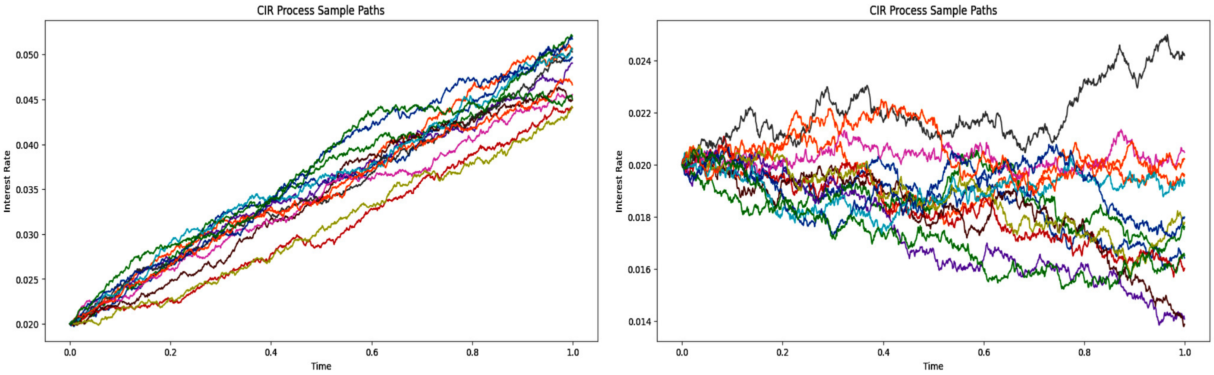
The mean-reversion speed parameter κ governs how quickly the process returns to its long-term mean. As such as it increases so does the rate at which the process reverts around it. The converse is also true as the following figure shows us:



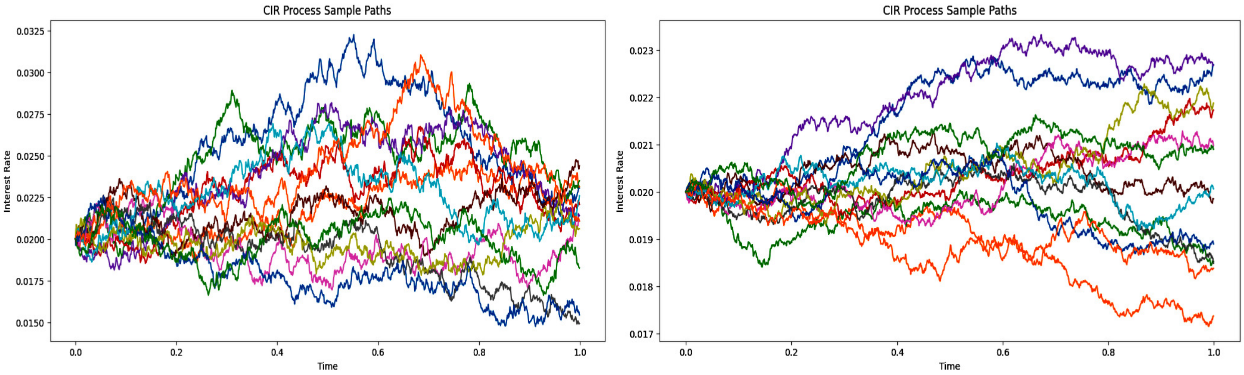
(a) The effect of increasing κ in the CIR process (b) The effect of decreasing κ in the CIR process

As we mentioned before, the long-term mean level parameter μ denotes the value towards which the process reverts. It acts as a catalyst for the process, and changing its value modifies the process's central tendency as we can see in the figure below:

The magnitude of random fluctuations or diffusion in the process is determined by the volatility parameter α. A higher level of volatility translates into higher deviations from the long-term mean, and vice versa.



(a) The effect of increasing μ in the CIR process (b) The effect of decreasing μ in the CIR process



(a) The effect of increasing α

(b) The effect of decreasing α

E Curve Finance

Curve Finance provides a one-of-a-kind trading mechanism by combining the ”constant sum automated market maker” (AMM) algorithm with the constant product formula. This unique technology allows for exact and predictable trading of stablecoins and equivalent assets while preserving market liquidity. The implementation of this combination technique by Curve provides good price discovery and stablecoin swaps with little slippage. In the paper- Mixing Constant Sum and Constant Product Market Makers by A. Port and N. Tiruviluamala we invite the reader to a careful reading of an indepth perspective of all the intricacies of a general AMM construction using this approach (see [11]). The Curve Finance improves traders’ and liquidity providers’ trading capabilities inside the decentralized finance ecosystem by combining the advantages of AMM with the constant product formula, as presented in the `whitepaper` (see [7]).

The value function for Curve Finance two token pools is given as follows

$$K_t D_t(x(t)+y(t)) + x(t)\,y(t) = K_t D_t^2 + \left(\frac{D_t}{2}\right)^2 . \tag{20}$$

Where, at time $t \geq t_0$, $x(t), y(t)$ represent the LP fraction of the pool total reserve respectively and $D_t = x(t) + y(t)$ denotes the LP total holdings. Considering A the amplification coefficient and the gamma factor $\gamma(t)$ we have that

$$K_0 = \frac{4\,x(t)\,y(t)}{D_t^2} \quad \text{and} \quad K_t = AK_0 \frac{\gamma^2(t)}{(\gamma(t)+1-K_0)^2} .$$

Evidently the entire liquidity pool's Invariant is achieved when we consider the current set of all the LP providing liquidity to that particular pool at time $t \geq t_0$.

E.1 *Understanding the Logic Underlying Curve's Expression of Crypto-Pools Value Function*

Curve Finance is optimized for stable swaps, which occur when a trader exchanges an amount of one stablecoin for another stablecoin, like DAI to USDC. Due to the nature of any stablecoin, in this type of trade one would expect to trade, for example, 1.000DAI to 1.000USDC, precisely.

Step 1

A value function that represents this relationship is known as Invariant Sum and it is given by

$$D_t = x(t) + y(t) .$$

On May 9th, 2022 Terra Luna stablecoin, UST, lost its peg to US Dollar, June 15th of the year 2023 the biggest stablecoin by market capitalization suffer a depeg to US Dollar and on February 17 of the year 2020 the Binance stablecoin, BUSD, traded below 0,96 US Dollar. This is to emphasize that depegging of stablecoins has been an usual event in the digital asset industry.

Step 2

To prevent the situation where one of the tokens is completely drained from a particular pool, the following Constant Product Invariant was added

$$x(t)y(t) = \left(\frac{D_t}{2}\right)^2$$

This invariant prevents the draining of a token in a given pool by increasing its price as the token's total reserve decreases in the pool. So we would have an Invariant that looks like

$$x(t) + y(t) + x(t)y(t) = D_t + \frac{D_t^2}{4}\,.$$

Step 3

Our main goal is to define an invariant that behaves like the constant sum when the token prices are stable, but like the constant product when they are not. As you may observe, we can achieve this by amplifying the constant sum term in the previous equation by a multiplicative constant, χ, resulting in the following invariant

$$\chi\big[x(t) + y(t)\big] + x(t)\,y(t) = \chi D_t + \frac{D_t^2}{4}\,.$$

So when $\chi = 0$ we have a Constant Product invariant governing the equation, and when $\chi \to \infty$ the Constant Product contribution in the overall invariant equation becomes infinitesimally small, consequently the Value Function behaves like an Invariant Sum which also occurs when the sum of $x(t) + y(t)$ is large enough. In order to arrive at an Invariant that behaves like Constant Product when $\chi = 0$ and behaves much like Invariant Sum when n $\chi \to \infty$ regardless of the how large or small $x(t) + y(t)$ is, we multiply the Invariant Sum equation by χD_t^{2-1} arriving at

$$\chi\big[x(t) + y(t)\big]D_t + x(t)\,y(t) = \chi D_t^2 + \frac{D_t^2}{4}\,,$$

which is precisely the StableSwap Invariant illustrated in the Curve Finance `whitepaper` (see [7]) for the two token pools.

Step 4

This value function can be further adapted as follows:

$$K_t D_t \big[x(t) + y(t)\big] + x(t)\, y(t) = K_t D_t^2 + \left(\frac{D_t}{2}\right)^2 \tag{21}$$

where we consider

$$K_0 = \frac{4\, x(t)\, y(t)}{D_t^2} \quad \text{and} \quad K_t = A K_0 \frac{\gamma^2(t)}{(\gamma(t) + 1 - K_0)^2}\,.$$

Following the `link` we have the illustration of this invariant when we vary the Amplification Coefficient $1 \leq A \leq 1.000$, the Invariant Sum constant, $15 \leq D_t \leq 70$, and the Gamma constant, $10^{-5} \leq \gamma \leq 10^{-3}$, considering $K_0 = 4x(t)y(t)/D_t^2$.

E.2 *IL Formula*

So far, we have derived the IL formula for the Balancer protocol, Uniswap V2 and V3 in the previous sections. As we were able to verify, the variation of token reserves was the sole factor that determines the pool's Value Function. In the case of Curve Finance however, beside these factors we have the Amplification coefficient, fee structures and the gamma constant that still influence the behavior of the pool. Naturally, in our discussion we will not be taking into consideration the fee structure of the Curve finance. Therefore, in order for us to work with a constant $K = K_t$ and a constant $D = D_t$ for each $t \geq t_0$, beside assuming no deposits nor withdrawals during the period the LP is providing liquidity in this protocol, we must also suppose that there is no change in the Gamma parameter present in the second page of the CurveCrypto `whitepaper`.

Notice that the gamma parameter would otherwise be updated through the Curve Finance governance process to rebalance incentives for that particular pool.

Let us consider the LP's token prices at instant $t \geq t_0$ as $p_{\mathrm{x}}(t), p_{\mathrm{y}}(t)$ and from this point on unless otherwise stated, we are going to denote the fraction of that particular pool reserve owned by the LP as $x(t)$ and $y(t)$ respectively in the same setting as we did in our discussion in previous sections about the Uniswap IL formula derivation.

Notice that the following property previously mentioned

$$p_{\mathrm{x}}(t) \cdot x(t) = p_{\mathrm{y}}(t) \cdot y(t)\,. \tag{22}$$

still holds true in the case of the Curve two tokens pool by exactly the same reasoning. As such we may re-write the value function described in (21) as:

$$KD\left(x(t) + x(t)\frac{p_{\mathrm{x}}(t)}{p_{\mathrm{y}}(t)}\right) + x^2(t)\frac{p_{\mathrm{x}}(t)}{p_{\mathrm{y}}(t)} = KD^2 + \left(\frac{D}{2}\right)^2,$$

as the reader may have noticed, we derived the previous equation by replacing $y(t)$ with its expression written in terms of $x(t)$ using the characterization described in equation 22.

Notice that, this results in a simple second degree polynomial equation that has the following solution:

$$x(t) = \frac{DK\,h(t)}{2\,p_{\mathrm{x}}(t)}.$$

where, in order to make our present discussion less cumbersome, we define the strictly positive valued function $h(\cdot)$ as:

$$h(t) := \sqrt{\left(p_{\mathrm{x}}(t) + p_{\mathrm{y}}(t)\right)^2 + p_{\mathrm{x}}(t)\;p_{\mathrm{y}}(t)\,(4/K + 1/K^2)} - p_{\mathrm{x}}(t) - p_{\mathrm{y}}(t)\,.$$

Analogously, we derive the solution in terms of the quantity $y(t)$ which yields:

$$y(t) = \frac{DK\,h(t)}{2\,p_{\mathrm{y}}(t)}$$

Hence, the amount invested by the LP at time t , i.e. $V_{\mathrm{invest}}(t)$ in dollar terms for $t \geq t_0$, will be given by

$$V_{\mathrm{invest}}(t) = x(t)\,p_{\mathrm{x}}(t) + y(t)\,p_{\mathrm{y}}(t) = DK\,h(t)\,.$$

Notice that, because we are working in the same setting as in the section of the Uniswap discussion, the quantities x, y, K and D only refer to the LP's holdings at time $t \geq t_0$, not to the total reserves of the entire pool. With that in mind, had the LP held their tokens instead, their holdings in dollars, at time $t > t_0$, which we will denote as $V_{\mathrm{hold}}(t)$, would be:

$$
\begin{aligned}
V_{\text{hold}}(t) &= p_{\text{x}}(t)\, x(t_0) + p_{\text{y}}\, y(t_0) = \Delta_{\text{x}}\, p_{\text{x}}(t_0)\, x(t_0) + \Delta_{\text{y}}\, p_{\text{y}}(t_0)\, y(t_0) \\
&= (\Delta_{\text{x}} + \Delta_{\text{y}})\, h(t_0) \frac{DK}{2}\,.
\end{aligned}
$$

Where we are using our usual notation to represent the quantities, i.e.;

$$
\Delta_{\text{x}} = \frac{p_{\text{x}}(t)}{p_{\text{x}}(t_0)}\,, \quad \Delta_{\text{y}} = \frac{p_{\text{y}}(t)}{p_{\text{y}}(t_0)}\,.
$$

Therefore, we may conclude that the IL the LP will face at time $t \geq t_0$ will be given by:

$$
\text{IL}(t) = \frac{V_{\text{invest}}(t) - V_{\text{hold}}(t)}{V_{\text{hold}}(t)} = \frac{2}{\Delta_{\text{x}} + \Delta_{\text{y}}} \frac{h(t)}{h(t_0)} - 1 \qquad \text{where}
$$

$$
h(t) = \sqrt{\big(p_{\text{x}}(t) + p_{\text{y}}(t)\big)^2 + p_{\text{x}}(t)\; p_{\text{y}}(t)\, (4/K + 1/K^2)} - p_{\text{x}}(t) - p_{\text{y}}(t)\,.
$$

References

[1] Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core, 2020. *URL: https://uniswap. org/whitepaper. pdf*, 2020.

[2] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. *Tech. rep., Uniswap, Tech. Rep.*, 2021.

[3] Andreas M Antonopoulos and Gavin Wood. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.

[4] M Antonopoulos Andreas. Mastering bitcoin: Programming the open blockchain, 2017.

[5] D Applebaum. Lectures on lévy processes, stochastic calculus and financial applications. *Ovronnaz, September*, 2005.

[6] David Applebaum. *Lévy processes and stochastic calculus*. Cambridge university press, 2009.

[7] Michael Egorov and Curve Finance. Automatic market-making with dynamic peg, 2021.

[8] Adrian Fischer, Robert E Gaunt, and Andrey Sarantsev. The variance-gamma distribution: A review. *arXiv preprint arXiv:2303.05615*, 2023.

[9] Dilip B Madan, Peter P Carr, and Eric C Chang. The variance gamma process and option pricing. *Review of Finance*, 2(1):79–105, 1998.

[10] Fernando Martinelli and Nikolai Mushegian. A non-custodial portfolio manager, liquidity provider, and price sensor. *URl: https://balancer. finance/whitepaper*, 2019.

[11] Alexander Port and Neelesh Tiruviluamala. Mixing constant sum and constant product market makers. *arXiv preprint arXiv:2203.12123*, 2022.

[12] Sheikh Munir Skh Saad and Raja Zahilah Raja Mohd Radzi. Comparative review of the blockchain consensus algorithm between proof of stake (pos) and delegated proof of stake (dpos). *International Journal of Innovative Computing*, 10(2), 2020.

[13] Wim Schoutens. *Lévy processes in finance: pricing financial derivatives*. Wiley Online Library, 2003.

[14] Neelesh Tiruviluamala, Alexander Port, and Erik Lewis. A general framework for impermanent loss in automated market makers. *arXiv preprint arXiv:2203.11352*, 2022.

[15] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.