Evaluating Fee Structures in Oracle Pricing Perpetual Futures Decentralized Exchanges: An Agent-Based Modelling Approach

Diogo Mendes Sampaio

Master's Dissertation

FEUP: Prof. Pedro Amorim
Ostium Labs: Kaledora Kierman-Linn

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Master in Industrial Engineering and Management

2024-06-27

Resumo

Esta tese investiga as complexidades das estruturas de taxas em *decentralized exchanges* de futuros perpétuos com preços baseados em *oracles*, utilizando uma abordagem de Simulação por Agentes. Explora como diferentes estruturas de taxas impactam o comportamento dos *traders*, a rentabilidade dos *liquidity providers* e a estabilidade geral da plataforma. O estudo destaca o papel das taxas dinâmicas de *price impact* em equilibrar o *open interest*. Através de simulações detalhadas que replicam os comportamentos de *trading*, a pesquisa fornece uma estrutura robusta para avaliar vários mecanismos de taxas. Além disso, a tese propõe uma nomenclatura para classificar as taxas, oferecendo uma abordagem padronizada para entender e comparar as estruturas de taxas em diferentes protocolos. As principais descobertas demonstram que os *arbitrageurs* e as estruturas de taxas dinâmicas aumentam significativamente a estabilidade da plataforma e reduzem o risco em comparação com as estruturas estáticas. A tese realça a importância de alinhar as taxas com as condições de mercado para participantes que reduzem melhorem a resiliência deste tipo de plataformas. Implicações práticas para o *design* de estruturas de taxas eficientes e estáveis são discutidas, juntamente com recomendações para pesquisas futuras sobre otimização das taxas e perfis diversos de *traders*. Este trabalho contribui para os esforços contínuos de refinamento dos *oracle pricing exchange models*, promovendo ecossistemas *decentralized finance* mais robustos e escaláveis.

Abstract

This thesis analyzes the complexity of fee structures in oracle pricing perpetual futures decentralized exchanges using an Agent-Based Simulation approach. It explores how different fee structures impact trader behavior, liquidity provider profitability, and overall platform stability. The study highlights the role of dynamic price impact fees in balancing open interest. Through detailed simulations that replicate trading behaviors, the research provides a robust framework for evaluating various fee mechanisms. Additionally, the thesis proposes a nomenclature to classify fees, offering a standardized approach to understanding and comparing fee structures across different protocols. Key findings demonstrate that arbitrageurs and dynamic fee structures significantly enhance platform stability and reduce risk compared to static structures. The thesis underscores the importance of aligning fees with market conditions to attract more participants and improve platform resilience. Practical implications for designing efficient and stable fee structures and recommendations for future research on fee optimization and diverse trader profiles are discussed. This work contributes to the ongoing efforts to refine oracle pricing exchange models, promoting more robust and scalable decentralized finance ecosystems.

Ackowledgments

I want to start by thanking my old friend Marco, who made it possible for me to work in a fast-paced ecosystem with brilliant people. Kaledora, thank you for all the support and attention you gave me throughout the small and big projects we tackled together. I also extend my gratitude to the entire team that did an amazing job building the Ostium app. Love you guys; this is just the warm-up.

Professor Pedro Amorim, thank you for your support and interest in helping me with a topic that was new to you. I hope you enjoyed learning more about it. A special thanks for your quick feedback near the end.

To my parents, who have always supported me in every decision along the way, thank you. None of this would have been possible without your unwavering support and encouragement.

Thanks to Salomé and the Turminha GPS for the unique moments we created together. University certainly wouldn't have been the same without you.

"The problem with experts is that they do not know what they do not know."

Nassim Taleb

Contents

1 Introduction **1**
1.1 DeFi and Perpetual Futures . . . 1
1.2 Oracle Pricing Model . . . 2
1.3 Problem Statement . . . 2
1.4 Dissertation Structure . . . 4

2 Background **5**
2.1 History of Money . . . 5
2.2 Decentralized Finance . . . 7
2.2.1 CeFi & DeFi . . . 7
2.2.2 DeFi Primitives . . . 8
2.2.3 On-Chain Asset Exchange . . . 9
2.2.4 CEX & DEX . . . 11
2.2.5 Derivatives . . . 11
2.3 Perpetual Futures . . . 12
2.3.1 Fundamental Concepts and Exchange Models . . . 14
2.4 Oracle Pricing Perpetual DEX . . . 16
2.4.1 Architecture . . . 16
2.4.2 Stakeholders . . . 17
2.4.3 Fees . . . 18
2.4.4 Risks and Mitigation Strategies . . . 19
2.4.5 Evaluation Metrics . . . 21
2.5 Gap in Background Literature . . . 22

3 Fee Structure Categorization **25**
3.1 Methodology . . . 25
3.2 Fee Categorization Nomenclature . . . 26
3.3 Order Execution . . . 27
3.4 Rolling . . . 28
3.5 Price Impact . . . 28
3.6 Looking for the optimal fee structure . . . 29

4 Methodology for Evaluating Fees **31**
4.1 Agent-Based Simulation . . . 31
4.1.1 Fundamentals of Agent-Based Simulation . . . 31
4.1.2 Implementation of Agent-Based Simulation in This Study . . . 32
4.2 Simulation Model Architecture . . . 33
4.3 Price . . . 33
4.3.1 Application in Simulation . . . 34

4.4 Trader . . . 35
4.4.1 Speculator . . . 35
4.4.2 Arbitrageur . . . 37
4.5 Trading Engine . . . 37
4.6 Fee . . . 39
4.7 Evaluation Criteria . . . 40
4.7.1 Value at Risk (VaR) . . . 41
4.7.2 Summary Statistics . . . 41
4.7.3 Statistical Tests . . . 42
4.8 Scenarios . . . 43
4.9 Example: Single Run . . . 43

5 Results **45**
5.1 Value at Risk . . . 45
5.2 PnL Breakdown . . . 45
5.2.1 PnL . . . 46
5.2.2 Gross PnL . . . 47
5.2.3 Fee . . . 48
5.2.4 OI Imbalance Score (ΔOI_{score}) . . . 49
5.3 Summary of finding . . . 50

6 Conclusion **51**

References **53**

A Background **55**

B Traders Algorithms **59**

C Fees Examples **63**
C.1 Order Execution Rates . . . 63
C.2 Rolling Rate . . . 63
C.3 Price Impact Rate . . . 64

D Simulation Model Parameters **65**

E Results Summary Metrics and Statistical Tests **69**

Acronyms and Symbols

ABS	Agent-Based Simulation
AMM	Automated Market Marker
APR	Annual Percentage Rate
CeFi	Centralized Finance
CEX	Centralized Exchange
CLOB	Central Limit Order Book
DAO	Decentralized Autonomous Organization
DeFi	Decentralized Finance
DEX	Decentralized Exchange
EVM	Ethereum Virtual Machine
GBM	Geometric Brownian Motion
LP	Liquidity Provider
MEV	Miner Extractable Value
OI	Open Interest
PnL	Profit and Loss
RWA	Real World Asset
TVL	Total Value Locked
VAMM	Virtual Automated Market Maker
VaR	Value at Risk

List of Figures

2.1 Background concepts hierarchy 5
2.2 Liquidity Pool dynamics in an AMM protocol by Aramonte et al. (2021) 10
2.3 Elements of exchange systems in Chen et al. (2024) 15
2.4 Oracle pricing model by Chen et al. (2024). In this diagram, diamonds represent on-chain smart contracts, rectangles represent off-chain databases, and ellipses represent off-chain servers. 17

4.1 Oracle Pricing Model simulation architecture 34
4.2 Fee Structure Charge-Temporal Representation 40
4.3 Vault and Traders Capital Flow 41
4.4 Example of single run with dynamic price impact fee and arbitrageurs 44

5.1 Value at Risk (VaR) with 90%, 95% and 99% Confidence Intervals for sum_PnL in Scenario 1, 2, and 3 46
5.2 PnL distribution by scneario 46
5.3 Gross PnL distribution by scenario 47
5.4 Fee distribution by scenario 48
5.5 Speculator fee distribution by scenario 49
5.6 Scenario 3 arbitrageur collected fee histogram 49
5.7 ΔOI_{score} by scenario 49

A.1 Cryptocurrency Monthly Exchange Volume by The Block (2024) 55
A.2 Bitcoin Monthly Futures Volume by The Block (2024) 56
A.3 Ethereum Monthly Futures Volume by The Block (2024) 56
A.4 DEX to CEX Futures Trade Volume by The Block (2024) 57
A.5 Opening a Perpetual Future in Binance Futures (2024) 57
A.6 Architecture diagram by Ostium Labs (2024b) 58

List of Tables

2.1 Comparison of different Perpetual Futures Exchange Models by Chen et al. (2024) 15

3.1 Categorization of fees in oracle pricing models 27
3.2 Order execution fees across different protocols 27
3.3 Rolling fees across different protocols 28
3.4 Price impact fees across different protocols 29

4.1 Trading Engine Variables 38
4.2 Trading Engine Formulas 39
4.3 Description of Position Data Frame Columns 42

D.1 Price Parameters 65
D.2 Speculator Parameters 66
D.3 Arbitrageur Parameters 66
D.4 Fee structure and Parameters 67
D.5 Specific-scenario parameters 67

E.1 Result sample mean and standard deviation 69
E.2 Statistical tests p-values 69

Chapter 1

Introduction

1.1 DeFi and Perpetual Futures

Decentralized Finance, commonly known as DeFi, represents a transformative shift in the financial sector, leveraging blockchain technology to create an open, permissionless, and highly transparent financial system. By eliminating intermediaries like banks and financial institutions, DeFi enables peer-to-peer transactions that empower users with full control over their assets. As of April 2022, the Total Value Locked (TVL) in DeFi systems was around \$150 billion, with Ethereum accounting for \$75 billion in TVL (Werner et al., 2022).

The rise of DeFi has led to the development of various financial primitives, including decentralized exchanges (DEXs), which facilitate the non-custodial exchange of digital assets. DEXs can be categorized based on their price discovery mechanisms into, i.e., order book DEXs and automated market makers (AMMs). AMMs, in particular, have gained popularity for their ease of liquidity provision and efficient, i.e., on-chain trading, allowing anyone to become a market maker and facilitating automatic rebalancing of liquidity pools (Barbon and Ranaldo, 2023).

One of the most significant innovations within DeFi is the introduction of perpetual futures. Conceptualized by Shiller (1993) and brought to the cryptocurrency market by BitMEX in 2016, perpetual futures deviate from traditional futures by eliminating the fixed delivery date, thereby enabling an indefinite trading period without the need for contract rollovers. This allows traders to maintain positions as long as they can cover the margin requirements, eliminating the complexities associated with contract expiry typical in traditional futures (Chen et al., 2024).

Despite their popularity, the majority of perpetual futures trading traditionally occurs on centralized platforms, raising concerns about the risks associated with centralization, such as platform security and counterparty risk. However, the rise of decentralized finance has begun to shift this landscape. Traders now increasingly engage in perpetual futures on decentralized platforms, leveraging innovations such as, i.e., the Hybrid Model, Oracle Pricing Model, and Virtual Automated Market Maker Model. Although decentralized exchanges currently handle a smaller volume compared to centralized exchanges, their growth rate surpasses that of their centralized counterparts, reflecting a growing preference for more transparent, secure, and inclusive trading environments (Chen et al., 2024).

1.2 Oracle Pricing Model

Oracle pricing DEX models have been an integral part of the DeFi landscape since their inception. The first notable oracle pricing DEX was Kwenta, launched in 2020. Since then, several other protocols, such as GMX and Gains Network, have adopted similar models. These early platforms set the standard for utilizing external price feeds to manage the trading of various assets. The common assets listed on these DEXs include cryptocurrencies like bitcoin and ethereum.

Among these platforms, Ostium stands out by listing new types of assets and creating a protocol specifically designed for Real World Assets (RWAs) such as gold, silver, and oil. By bridging the gap between traditional financial markets and decentralized finance, Ostium enables traders to engage in leveraged trading of commodities and foreign currencies directly on a decentralized platform, permissionlessly. The inclusion of RWAs and high-leverage trading requires an oracle pricing model to provide an accurate price of the underlying asset, regardless of the volume on the exchange.

This space continues to grow as new protocols are deployed and existing ones are continuously upgraded. The dynamic nature of oracle pricing DEXs ensures that they remain at the forefront of innovation within DeFi, constantly evolving to meet the demands of traders and the complexities of global financial markets. As these platforms enhance their features and expand their listed assets, they contribute to the robustness and resilience of the decentralized financial ecosystem.

1.3 Problem Statement

Oracle Pricing Perpetual Futures DEXs represent a unique category of decentralized exchanges that fetch prices from external oracles rather than relying on internal price discovery mechanisms like the Central Limit Order Book (CLOB) model. In a CLOB model, traders define prices through their buy and sell orders, requiring direct counterparties for every trade. In contrast, oracle pricing models use external entities to provide real-time prices for derivatives, enabling asynchronous, on-chain trade execution without the need for an immediate counterparty (Ostium Labs, 2024a).

Despite their shared foundational principles, these protocols often use different nomenclature for similar concepts and implement varying internal mechanisms as they expand and upgrade to new versions. This lack of standardization poses significant challenges. For users to fully understand the fee structures, they must go to detailed documentation or directly reading the smart contract code, which can be cumbersome and non-intuitive.

One of the primary challenges with oracle pricing models is counterparty risk. This risk arises from the protocol's obligation to fulfill user profits, which can strain the liquidity pool if not managed properly. In traditional finance, Contracts for Difference (CFD) offer a somewhat analogous situation (Leelarthaepin, 2006), though documentation on their dynamic fee and spread adjustments is sparse. Unlike traditional finance, DeFi protocols are generally more transparent, disclosing detailed information about their fee structures and risk management strategies. However, there must be clear incentives for traders to understand and mitigate the risks that the protocol faces, emphasizing the importance of transparent fee formulas (Werner et al., 2022).

Currently, there is no systematic categorization of fee structures across Oracle Pricing Perpetual Futures DEXs. This lack of standardization complicates the user experience, as traders must navigate different terminologies and understand distinct fee mechanisms to trade efficiently across multiple platforms. Transparency and standardization are fundamental to DeFi, enabling users to seamlessly switch between DEXs and clearly understand the fees being charged.

Furthermore, it is not clear which fee structure is the best one. These exchanges use different combinations of fee structures and parameters, making it difficult to assess which fee architecture is the most capital-efficient. While some companies, such as Chaos Labs, have produced valuable reports on the optimization of fees and parameters for oracle pricing protocols, there remains a gap in comprehensive, comparative analysis.

Therefore, the two primary problems at hand are:

1. Lack of Systematic Categorization of Fee Structures: there is no standardized nomenclature or categorization for the various fee structures used by different Oracle Pricing Perpetual Futures DEXs. This inconsistency makes it challenging for users to understand and compare fees across different platforms;

2. Uncertainty of Optimal Fee Structure: it is unclear which combination of fee structures and parameters provides the best capital efficiency and risk management for Oracle Pricing Models. This uncertainty requires further research and analysis to determine the most effective fee architecture.

To address the first challenge, a comprehensive comparison between the fee structures of GMX, gTrade, and Avantis will be performed. This analysis will involve collecting detailed fee information from each protocol and categorizing these fees based on the moment they are charged and the specific trading component they affect. By systematically organizing fees according to their timing and impact on trading activities, we can elucidate the main types of fees and their functions within each protocol.

Secondly, to systematically analyze and compare different protocol structures, a quantitative model will be designed to simulate the trading engines used by oracle pricing models, incorporating the modular fee structures identified in the fee categorization chapter 4. This simulation will leverage Agent-Based Simulation, where traders are represented as agents to more accurately mimic real-world trading behaviors. Monte Carlo simulations will be employed to compute multiple price paths using geometric Brownian motion. This approach generates a large number of samples, thereby increasing the precision in estimating evaluation parameters. By replicating the intricate interactions and price fluctuations within the trading environment, this quantitative model aims to provide a robust framework for evaluating the efficiency and effectiveness of various fees.

1.4 Dissertation Structure

Chapter 2 of this master's thesis provides a comprehensive overview of the suggested background to understand the topics discussed in the upcoming chapters.

In Chapter 3, a nomenclature to categorize fees across exchanges is proposed.

Addressing the second problem, Chapter 4 describes the methodology used to analyze the performance of fees. Following this, Chapter 5 presents the results obtained by applying the previous method in different scenarios.

Finally, Chapter 6 presents the study's main conclusions, highlighting the key findings and their significance. Additionally, the limitations of the present study and potential areas for future research will be outlined.

Chapter 2

Background

This chapter begins with a brief overview of the history of money, emphasizing the technological advancements that have been crucial in defining value and facilitating its transfer. Following this, the focus shifts to DeFi, exploring its various components including asset exchange and derivatives. A special emphasis is placed on Oracle Pricing Perpetual Futures Models. Figure 2.1 illustrates the relationships and hierarchy between these key concepts, providing a clear visual representation of the chapter's content hierarchy.

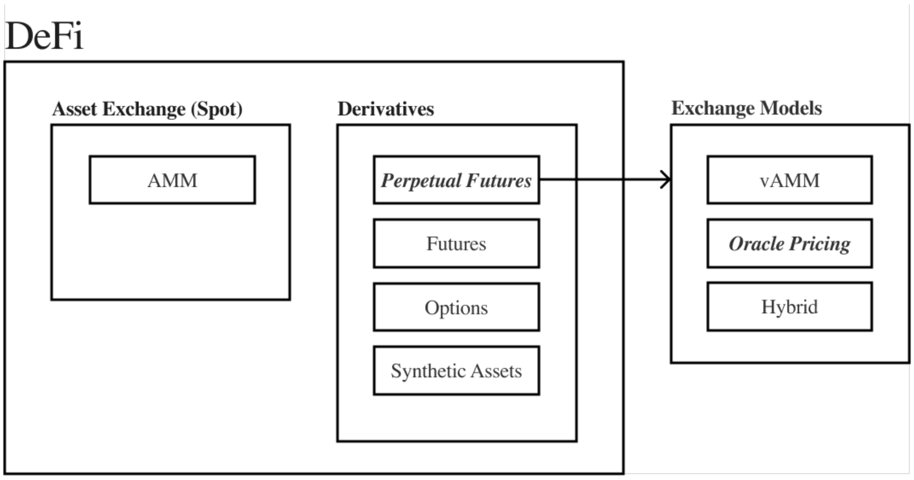


Figure 2.1: Background concepts hierarchy

2.1 History of Money

The history of money dates back thousands of years ago, evolving from simple barter systems to sophisticated digital currencies. This evolution reflects not just changes in economic activities but also advancements in societies, technologies, and global interactions (Bailey, 1995). This subsection delves into key milestones in the development of money, tracing its transformation from tangible assets to digital forms.

Before the advent of money, people used the barter system, where goods or services were exchanged for goods or services directly. In this method of exchange, individuals or entities negotiate and exchange items based on their perceived value to each other. However, limitations of barter systems such as double coincidence of wants, low durability, and difficult portability, led to the adoption of commodity money. Commodity money is a type of currency with intrinsic value such as metals (gold, silver), shells, and livestock. This adoption created a more acceptable, flexible, and efficient medium of exchange (Bailey, 1995).

As trade expanded, the need for a more convenient and standardized form of money led to the usage of metal coins. The invention of coinage (design, minting, and distribution of coins) is attributed to the ancient kingdom of Lydia (modern Turkey) in the 6th BCE, creating the first standardized coin made of electrum, a natural alloy of gold and silver. The concept of coinage quickly spread to other civilizations, including Greece, Rome, China, and India. Therefore, the development of coinage marked a significant advancement in the history of money, providing a more efficient and standardized medium of exchange compared to earlier forms of exchange (Bailey, 1995).

China first introduced fiat paper money in the ninth century CE, where private merchants began issuing "exchange bills" (known as *jiaozi*). These bills grew in circulation but were prone to abuse, leading to a surge in legal disputes. The success of fiat paper money depended on government control and large-scale operations to unify the territory (Peneder, 2022). This innovative form of currency quickly spread to European territories, offering significant advantages over metal coins, such as being lightweight, easy to transport, and more convenient for large transactions. Over time, the backing of paper money shifted from precious metals to the authority and trust in the issuing government or financial institution, resulting in fiat currencies (Bailey, 1995). For instance, in the United States, banknotes were backed by a specific amount of gold as established by the Federal Reserve until 1971, when the gold standard was completely replaced by fiat money (Lioudis, 2024).

The financial sector was among the first to adopt new information and communication technologies in the second half of the twentieth century. Thus, money and banking co-evolved with industrial technology, leading to the evolution of digital payment cards, electronic banking, and mobile payments. Amid the financial crisis of 2008-09, cryptocurrencies disrupted the money institutions with the introduction of the Bitcoin protocol (Nakamoto, 2008). It established a peer-to-peer system of electronic cash that operates independently, without the intermediate of a financial organization (Peneder, 2022).

The history of money reflects humanity's ingenuity in adapting to economic needs and technological advances (Bailey, 1995). From barter to blockchain, the evolution of money continues to shape the way we conduct trade and organize our societies, promising further innovation and transformation in the future.

2.2 Decentralized Finance

Decentralized Finance, commonly known as DeFi, refers to a blockchain-powered peer-to-peer financial system that eliminates intermediaries like banks and financial institutions. DeFi leverages blockchain technology to create a new financial architecture that is non-custodial, permissionless, and openly auditable. This ecosystem allows users to maintain full control over their funds, interact with financial services without censorship, audit the system's health, and create new financial products and services from existing ones. As of April 2022, the total value locked (TVL) in DeFi systems was around $150 billion, with Ethereum accounting for $75 billion in TVL (Werner et al., 2022). At the moment, in June 2024, that value is smaller, the DeFi TVL is around $100 billion, and Ethereum accounts for $60 billion, data from DeFiLlama (2024).

2.2.1 CeFi & DeFi

Centralized Finance (CeFi) and Decentralized Finance (DeFi) represent two distinct paradigms in the financial world, each with its own advantages, challenges, and implications for the future of financial systems. CeFi encompasses the traditional financial ecosystem where centralized institutions like banks, insurance companies, and regulatory bodies manage financial transactions and services. These institutions are responsible for maintaining the stability, security, and efficiency of financial markets. They oversee the issuance and management of fiat currencies, provide essential services such as lending, borrowing, and investing, and enforce regulatory compliance. The centralized control in CeFi ensures trust and security, but it also leads to inefficiencies, higher costs, and limited access for certain populations due to the reliance on intermediaries.

In contrast, DeFi leverages blockchain technology to offer a decentralized alternative, aiming to provide financial services without the need for traditional intermediaries. DeFi utilizes smart contracts, which are self-executing contracts with the terms of the agreement written directly into code, enabling automated, transparent, and secure transactions. This decentralized approach allows users to retain control over their assets, reducing the need for trust in intermediaries and offering greater transparency and accessibility. However, the decentralized nature of DeFi introduces challenges such as regulatory uncertainties, security vulnerabilities, and the need for widespread user education.

The relationship between CeFi and DeFi is increasingly synergistic, with both systems influencing. Traditional financial institutions are exploring DeFi technologies to enhance their services and remain competitive, while many DeFi protocols rely on CeFi elements such as stablecoins pegged to fiat currencies and price feeds from centralized exchanges. This integration is fostering a more robust and inclusive financial ecosystem, leveraging the strengths of both paradigms.

The primary differences between CeFi and DeFi lie in their operational structures and underlying philosophies. CeFi's centralized control ensures regulatory compliance and stability but can lead to inefficiencies and higher costs. In contrast, DeFi's decentralized network eliminates intermediaries, offering greater transparency and efficiency but also posing significant regulatory and security challenges. CeFi institutions have significant control over users' assets and data, potentially leading to issues of privacy and trust. DeFi, with its blockchain foundation, provides transparent and immutable records of transactions, reducing the need for trusted intermediaries but exposing users to privacy risks.

Financial services in CeFi are backed by regulatory frameworks and insurance protections, while DeFi replicates these services at lower costs and higher yields due to the absence of intermediaries. As CeFi and DeFi continue to evolve and integrate, they are likely to create a more dynamic, efficient, and inclusive financial ecosystem, combining the stability and regulatory oversight of CeFi with the transparency, accessibility, and innovation of DeFi (Qin et al., 2021).

2.2.2 DeFi Primitives

Blockchain serves as the backbone of decentralization. DeFi applications are built using smart contracts, which operate autonomously on the blockchain. These applications often utilize keepers and oracles to monitor and interact with off-chain data, ensuring the protocol functions correctly. Governance tokens empower the community to vote on and dictate protocol upgrades, promoting a decentralized and democratic decision-making process (Werner et al., 2022).

Blockchain technology underpins the entire DeFi ecosystem. It operates as a transaction-based state machine, where each node in the network maintains a synchronized copy of the ledger. This decentralized ledger ensures transparency, security, and immutability of transactions. When a network participant initiates a transaction, it is broadcast to the network, validated by peers, and stored in a mempool. Miners then select transactions from the mempool to include in the next block, often prioritizing those with higher transaction fees. The sequential execution of transactions within blocks ensures the integrity and consistency of the blockchain (Werner et al., 2022). As stated by Nico (2023) 'A blockchain is a public database that is updated and shared across many computers in a network'.

Werner et al. (2022) identify the following blockchain layers that are most relevant to DeFi:

- Smart Contracts and Transactions: smart contracts are programmable, self-executing contracts that reside on the blockchain. They interact with one another within the same execution context through message-calls and are designed to support atomicity, ensuring transactions either fully succeed or entirely fail without leaving any partial states. The Ethereum Virtual Machine (EVM) is a prime example, operating as a stack machine using gas to measure computational effort. Smart contracts must be expressive enough to encode complex protocol rules, support conditional execution and bounded iteration, and allow for inter-contract communication to maintain composability. Transactions are initiated by network participants and broadcast to peers, where they are validated and stored in a mempool. Miners then include these transactions in blocks, executing them sequentially. The order of execution is determined by the miner, who may reorder transactions to maximize revenue, a practice known as Miner Extractable Value (MEV). This process highlights the role of miners in maintaining blockchain integrity while also introducing potential manipulation risks;

- Keepers: external entities incentivized to trigger necessary state updates in DeFi protocols. Since smart contracts cannot autonomously initiate transactions, they rely on keepers to maintain protocol functionality. For example, protocols may require automatic liquidation of user collateral under specific conditions, prompting keepers to initiate the corresponding transactions. This ensures that the protocol's on-chain state remains accurate and secure, enabling continuous and reliable operation;

- Oracles: the bridge between off-chain data and blockchain smart contracts, enabling the integration of external information such as asset prices or event outcomes. They vary in design and risk profiles, with centralized oracles relying on single trusted data providers and decentralized oracles distributing data reporting across multiple participants. Decentralized oracles incentivize accurate reporting to mitigate trust issues, though they also face their own challenges. The integrity of oracles is crucial for the accurate functioning of DeFi protocols, as incorrect data can lead to erroneous contract execution and financial losses;

- Governance: a mechanism for evolving and adjusting protocol parameters. Initially, many DeFi protocols start with centralized governance under a benevolent dictator or a multisig council, with plans to transition to decentralized governance. This is typically achieved through governance tokens, which grant holders the right to propose and vote on protocol changes. Governance tokens represent ownership in a Decentralized Autonomous Organization (DAO), responsible for the protocol's stewardship. Proposals for protocol upgrades are submitted as executable code, and a quorum of votes is required for enactment. This ensures that the protocol evolves in a way that reflects the collective interests of the token holders, promoting a decentralized and democratic decision-making process.

2.2.3 On-Chain Asset Exchange

Decentralized exchanges (DEXs) are a pivotal class of DeFi protocols that enable the non-custodial exchange of digital assets, with all trades settled on-chain and thus publicly verifiable. Initially, DEXs supported only assets native to their respective blockchains. However, the introduction of wrapped tokens, such as wBTC (wrapped Bitcoin), and advanced cross-chain solutions have allowed DEXs to expand beyond these limitations (Werner et al., 2022).

DEXs can be categorized based on their price discovery mechanisms into order book DEXs and automated market makers (AMMs). Order book DEXs further divide into individual and batch settlement systems. However, due to their widespread adoption and unique characteristics in DeFi, AMMs are the primary focus of this discussion. Figure 2.2, by Aramonte et al. (2021), illustrates the AMM dynamics.

In traditional finance, market makers provide liquidity by quoting bid and ask prices and profiting from the bid-ask spread. These strategies can become highly sophisticated optimization

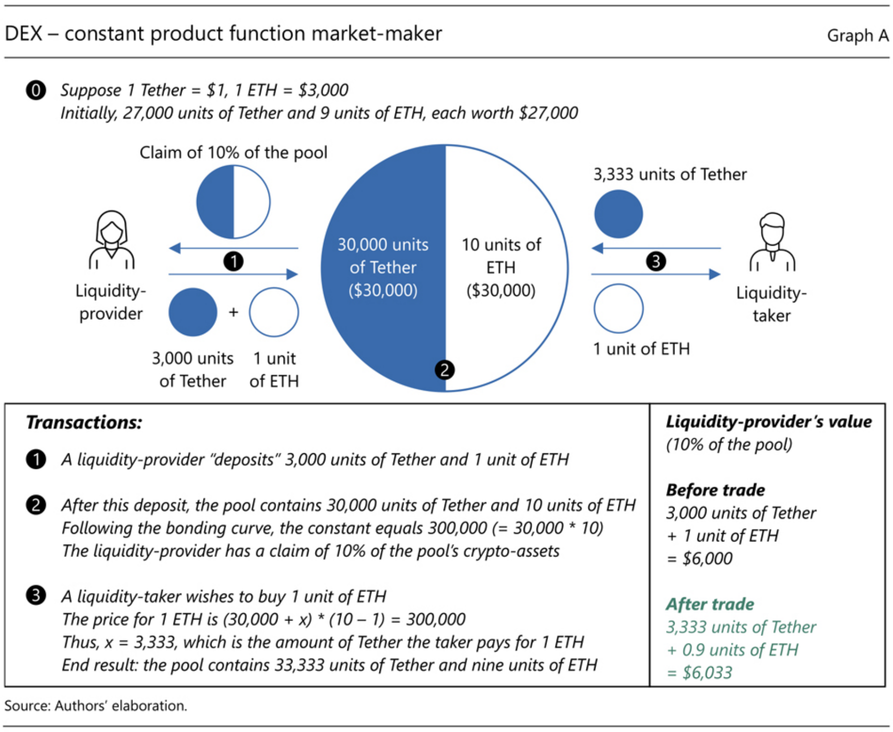


Figure 2.2: Liquidity Pool dynamics in an AMM protocol by Aramonte et al. (2021)

problems. In contrast, AMMs use algorithmic methods to provide liquidity through simple pricing rules, replacing order books with on-chain liquidity pools. Previously studied in algorithmic game theory, such as the logarithmic market scoring rule (LMSR) used in prediction markets, AMMs have gained popularity in DeFi due to factor highlighted by Werner et al. (2022):

- Ease of Liquidity Provision: AMMs facilitate the provision of liquidity for minor assets;
- Accessibility: anyone can become a market maker, even if returns are suboptimal;
- Automatic Rebalancing: AMM pools can function as automatically rebalancing portfolios;
- Efficiency: maintaining an order book on-chain is inefficient, whereas AMMs offer a streamlined alternative.

In an AMM liquidity pool, reserves for two or more assets are locked into a smart contract. Liquidity providers receive newly minted liquidity tokens representing their share of the pool. Trades occur by exchanging against the smart contract's liquidity reserves, adding to one token's reserves while withdrawing from others. Trading fees are retained by the liquidity pool and distributed proportionally to liquidity token holders. To redeem their share of liquidity and accrued fees, providers must return their liquidity tokens (Werner et al., 2022).

AMMs determine asset prices using a deterministic formula based on the relative sizes of the liquidity on each side of a currency pair. Thin liquidity can cause significant price fluctuations, creating arbitrage opportunities. Arbitrageurs buy or sell the same asset across different markets to profit from price differences, ensuring AMM prices align with the open market. However, changing reserve ratios mean that liquidity providers may withdraw different token ratios compared to their initial deposits Werner et al. (2022).

2.2.4 CEX & DEX

Centralized exchanges (CEXs) and decentralized exchanges (DEXs) represent two distinct models in the cryptocurrency trading ecosystem. CEXs like Binance, Kraken, and Coinbase operate under a centralized model where a single organization manages the platform. They use a central limit order book (CLOB) system, matching buy and sell orders to facilitate trades, similar to traditional stock exchanges. CEXs hold users' funds in custodial wallets, offering user-friendly interfaces, robust customer support, and regulatory compliance, which provide a sense of security and legal recourse. However, this centralization introduces risks such as potential hacks, mismanagement, and custodian risks, as evidenced by incidents like the collapse of FTX (Barbon and Ranaldo, 2023).

In contrast, DEXs like Uniswap, Sushiswap, and PancakeSwap operate on blockchain technology, enabling peer-to-peer trading without a central authority. DEXs use AMMs and liquidity pools managed by smart contracts, allowing users to maintain control over their assets by interacting directly from their wallets. This approach offers greater transparency, security against centralized failures, and access to innovative financial products like yield farming. However, DEXs face challenges such as higher transaction costs due to gas fees (transaction costs), potential smart contract vulnerabilities, and a steeper learning curve for users unfamiliar with blockchain technology (Barbon and Ranaldo, 2023).

CEXs and DEXs illustrate the broader financial ecosystem of CeFi and DeFi. Both aim to facilitate efficient and secure trading but differ fundamentally in their operational models. CEXs control users' funds and adhere to regulatory standards, providing stability and security but facing custodian and centralization risks. DEXs are non-custodial, operate in a less regulated environment, and rely on AMMs, offering transparency and innovation but introducing smart contract risks (Barbon and Ranaldo, 2023).

2.2.5 Derivatives

Derivatives are financial contracts whose value is derived from the performance of underlying assets. As of March 2022, the derivatives market accounts for approximately 62% of the entire cryptoassets trading market. Although about 99% of derivative trading volume occurs on centralized exchanges, several DeFi protocols have emerged to offer similar functionalities (Werner et al., 2022). Currently, derivative trading volume on decentralized exchanges increased to 15%, data from DeFiLlama (2024) and The Block (2024).

These protocols specialize in synthetic assets, futures, perpetual swaps, and options. According to Werner et al. (2022), the four fundamental types of derivatives are:

1. Synthetic Assets: in DeFi, synthetic assets typically replicate off-chain assets on-chain, such as the USD in protocols like Maker and Synthetix. Another mechanism for constructing synthetic assets involves using AMMs that enact dynamic portfolio rebalancing strategies to replicate derivative payoffs. These strategies are similar to synthetic portfolio insurance found in traditional finance and have been explored using constant product market makers;

2. Future: have seen limited adoption in DeFi, primarily due to the high volatility of underlying cryptoassets, which makes it challenging to determine the risk taken by traders writing futures contracts. This volatility introduces significant uncertainty and risk for both traders and platforms;

3. Perpetual Futures (or Perpetual Swaps): similar to futures but do not have an expiry date or settlement and were specifically created for cryptoasset markets. These instruments allow traders to maintain positions by providing funding transactions if their positions are underfunded. Due to frequent price discovery, the price of perpetual swaps tends to trade closer to the underlying asset's price compared to futures. Additionally, perpetual futures are more capital-efficient than trading the underlying asset itself, as platforms require less than 100% collateral to be posted by traders;

4. Options: the DeFi market for options is still in its early stages, with basic call and put options available . The limited adoption of options in DeFi can be attributed to three main factors. First, current option platforms require at least 100% collateralization, which is less capital-efficient compared to their centralized counterparts. Second, derivatives with set expiry dates (futures or options) are difficult to price on AMMs. Most of these platforms do not account for a time dimension in the asset, causing an issue with option trading since the value of the option is subject to time decay. Third, options require a liquid market for efficient price discovery. Addressing the mentioned issues is essential to bootstrap the necessary liquidity for efficient option pricing.

2.3 Perpetual Futures

In this section, perpetual futures are introduced as a novel financial instrument that allows traders to gain exposure to underlying assets without the constraints of an expiration date. Additionally, this section identifies and discusses the main types of exchange models used for perpetual futures, highlighting their distinct mechanisms and operational differences.

Perpetual futures, conceptualized by Shiller (1993), represent a significant innovation in financial derivatives. Introduced into the cryptocurrency market by BitMEX in 2016, these instruments deviate from traditional futures by eliminating the fixed delivery date, thereby enabling an indefinite trading period without the need for contract rollovers. This innovation allows traders to maintain positions as long as they can cover the margin requirements, eliminating the constraints and complexities associated with contract expiry typical in traditional futures (Chen et al., 2024).

Perpetual futures introduce a unique mechanism known as funding fees to ensure that the price of the futures contracts remains aligned with the underlying spot price. Unlike traditional futures, where the price difference or basis between the futures and spot prices narrows as the contract approaches expiration due to arbitrage activities, perpetual futures use these funding fees to adjust the price dynamically. Depending on market conditions, these fees are periodically exchanged between long and short traders (Chen et al., 2024).

Another distinctive feature of perpetual futures is the high leverage they offer, significantly higher than that available in traditional futures markets. Platforms like BitMEX offer leverages up to 100x for BTC-USD trading pairs, with some platforms offering even more aggressive leverage like Gains Network. This high leverage availability has made perpetual futures a preferred instrument for traders seeking substantial exposure with minimal capital outlay (Chen et al., 2024). Figures A.1, A.2, and A.3 in Appendix A show the trading volume for Bitcoin perpetual futures reached impressive heights in 2021 and 2022, overlooking volumes seen in other derivatives like futures and options.

The majority of perpetual futures trading traditionally occurs on centralized platforms as illustrated in Figure A.4, Appendix A, that only 1%-2% of perpetual futures trading occurs on-chain. This rases concerns about the risks associated with centralization such as platform security and counterparty risk like the FTX collapse incident (Reiff et al., 2024). However, the rise of DeFi has begun to shift this landscape. Traders now increasingly engage in perpetual futures on decentralized platforms, leveraging innovations such as the Hybrid Model, Oracle Pricing Model, and Virtual Automated Market Maker Model (VAMM). Although decentralized exchanges currently handle a smaller volume compared to centralized exchanges, their growth rate reflects a growing preference for more transparent, secure, and inclusive trading environments (Werner et al., 2022).

The main features of perpetual futures, as presented by Hayes et al. (2023), include:

- No expiration date: the primary feature of perpetual futures is the absence of an expiration date, allowing traders to keep their positions open indefinitely without the need to close or roll over their contracts. Rolling over a futures contract involves switching from the front month contract, which is close to expiration, to another contract with a later expiration date. This process is typically done to avoid the costs and obligations associated with the settlement of expiring contracts (Lioudis et al., 2021);

- Funding Rate: perpetual futures use a mechanism funding rate to keep the price close to the underlying asset's spot market price. One side of the contract pays this rate to the other, depending on the difference between the perpetual futures price and the spot price. For example, if the spot price is above perpetual, longs pay funding to shorts. If the spot price is below perpetual, shorts pay funding to longs;

- Leverage: leverage can amplify profits and losses, allowing traders to control larger positions with smaller amounts of capital;

- Margin Requirements: traders need to maintain a minimum margin balance. If the balance falls below the maintenance margin requirements, the position is closed to prevent further losses, also known as liquidation.

To open a position on the Binance platform for a perpetual futures contract, several key inputs are required. Figure A.5 in Appendix A illustrates the Binance Futures interface. First, the underlying asset must be selected, which is the asset associated with the contract. Next, the position type must be chosen: buy/long, which profits from price increases, or short/sell, which profits from price decreases. The opening price is the price at which the contract is initially bought or sold. Leverage must also be specified, as it sets the required initial and maintenance margins. Higher leverage allows traders to control larger positions with less capital but also increases risk. The size of the position, which is the number of units of the trading instrument being bought or sold, must be defined. The initial margin or collateral, which is the required capital to open the position, is specified in terms of the base or quote asset (for example, in the BTC/USDT pair, BTC is the base asset and USDT, a stablecoin, is the quote asset). The maintenance margin is the required capital to prevent liquidation; if the margin ratio falls below this threshold, the position will be liquidated to prevent further losses. The funding rate, which is influenced by the asset's index and mark price, must also be considered. Finally, the closing price is the price at which the contract is closed, determining the PnL of the position.

For example, consider opening a long position on the BTC/USDT pair with an opening price of \$50,000. Using leverage of 10x, the total size of the position is \$5,000, and the initial margin required is \$500. While the position is open, if the funding rate is positive, it charges \$5 to the position. The trader decides to close the position at a price of \$51,000, resulting in a profit of \$95 after the funding rate.

2.3.1 Fundamental Concepts and Exchange Models

Perpetual Futures exchanges use different models, as it was identified by Chen et al. (2024). The key stakeholders and elements are represented in Figure 2.3 and outlined as follows:

- Traders: individuals or organizations actively involved in buying and selling perpetual contracts. They provide collateral to manage and maintain their positions effectively throughout the trading process;

- Custody Module: this module safeguards the assets held in all trader accounts. It continuously updates to reflect the most current balance information and supports the execution of both deposits and withdrawals by traders;

- Matching Module: tasked with storing, matching, and executing buy and sell orders for contracts, this module ensures efficient transaction processing;

- Risk Control Module: crucial for monitoring and evaluating each trader's position based on executed orders, this module ensures that the collateral provided is adequate to cover potential losses. In certain cases, it takes over a trader's position to initiate liquidation procedures.

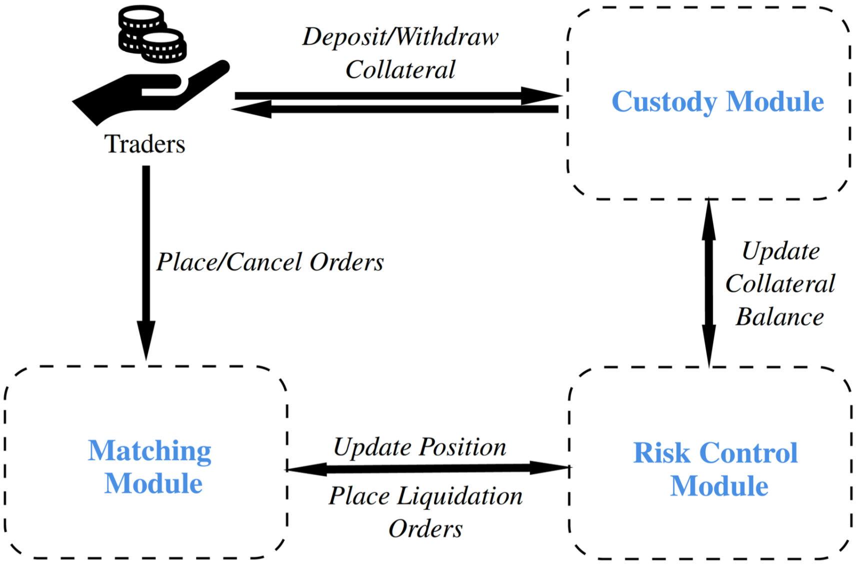


Figure 2.3: Elements of exchange systems in Chen et al. (2024)

Table 2.1 provides a detailed horizontal comparison of various perpetual futures exchange models across five key dimensions: Custody, Matching, Risk Control, Counterparty, and Pricing. The Centralized Exchange Model employs a combination of hot and cold wallets for asset custody, off-chain servers for trade matching and risk control, with traders acting as counterparties and pricing determined by trader activity. The Hybrid Model retains off-chain servers for matching and risk control while shifting custody to smart contracts, thus introducing a layer of decentralization. In both CEX and Hybrid models, the centralized control of certain elements ensures high efficiency and aligns closely with traditional financial practices.

Table 2.1: Comparison of different Perpetual Futures Exchange Models by Chen et al. (2024)

Type	**Custody**	**Matching**	**Risk Control**	**Counterparty**	**Pricing**
CEX	Hot & Cold Wallet	Off-Chain Server	Off-Chain Server	Traders	Traders
Hybrid	Smart Contract	Off-Chain Server	Off-Chain Server	Traders	Traders
Oracle Pricing	Smart Contract	Smart Contract	Off-Chain Keepers	Liquidity Providers	Oracle
VAMM	Smart Contract	Smart Contract	Off-Chain Keepers	Liquidity Providers	AMM Algorithm

In contrast, the Oracle Pricing and Virtual Automated Market Maker models lean heavily on decentralization, utilizing smart contracts for both custody and trade matching. These models rely on liquidity providers as counterparties, with pricing determined by external oracles and automated algorithms, respectively. Despite their decentralized nature, these models still incorporate centralized components for risk control, such as off-chain keepers and oracles. This mixed approach aims to balance the benefits of decentralization with the necessity of efficient risk management. As shown in the table, the Oracle Pricing model utilizes an oracle for precise price determination, while the VAMM model employs an AMM algorithm.

2.4 Oracle Pricing Perpetual DEX

The problems stated in Chapter 1 are specifically related to Oracle Pricing Perpetual DEXs. This section highlights the architecture and stakeholders in this model, providing a brief introduction to the associated fees, outlining the risks and mitigation strategies, and performance evaluation metrics unique to this exchange model.

2.4.1 Architecture

The oracle pricing model represents a sophisticated approach within cryptocurrency exchanges, strategically balancing decentralized and centralized components to enhance security, transparency, and efficiency. This model's architecture is designed to leverage the benefits of on-chain smart contracts for key functions such as fund custody, trade execution, and storage, while still relying on centralized elements for specific tasks like risk control and trade price confirmation. Figure 2.4 illustrates the key elements and interactions that are specific to the oracle pricing model.

Central to the Oracle Pricing Model are two categories of participants: traders and liquidity providers. Traders actively place orders through the exchange's interface or directly via on-chain smart contracts, similar to participants in centralized and hybrid exchange models. Liquidity providers passively accept orders, acting as counterparties by depositing funds into an on-chain Liquidity Pool, ensuring a dynamic and liquid trading environment.

The custody of funds is managed by two on-chain smart contracts. The collateral custody smart contract safeguards traders' collateral, maintains position balances, and settles profits and losses, ensuring secure and accurate asset management. The liquidity custody smart contract manages the funds of liquidity providers, calculating and storing their balances, which is essential for maintaining the integrity and security of the liquidity pool.

Trade execution is handled by the trading smart contract, which executes trades and updates position statuses. This use of smart contracts ensures secure and transparent transactions, reducing the potential for manipulation and enhancing participant trust.

The oracle pricing model still relies on some centralized components for risk control. The price oracle network provides real-time asset prices to the trading smart contract, ensuring accurate trade execution and preventing price manipulation. Keepers, off-chain servers, monitor positions and handle liquidations when necessary, maintaining market stability. Additionally, the trading smart

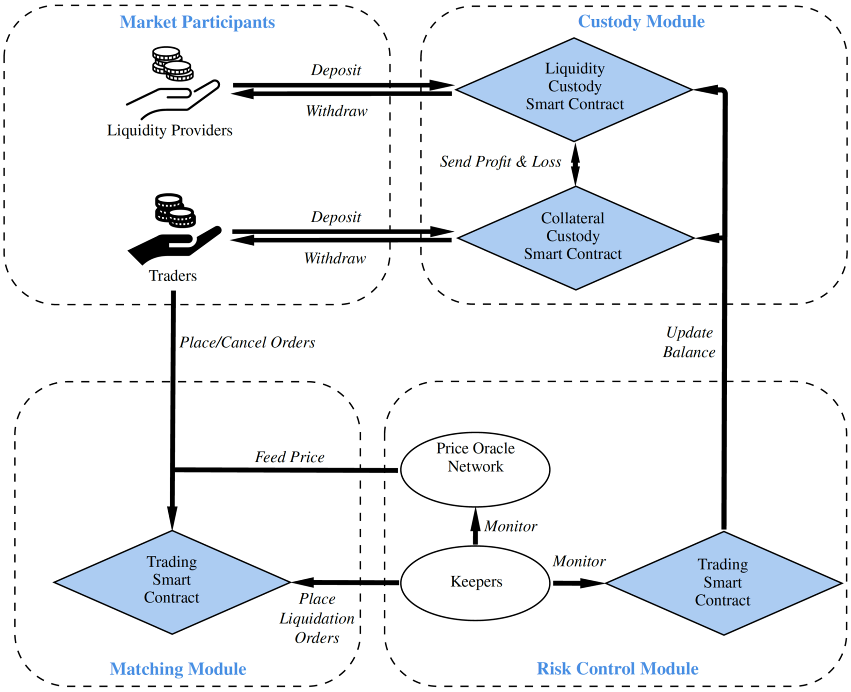


Figure 2.4: Oracle pricing model by Chen et al. (2024). In this diagram, diamonds represent on-chain smart contracts, rectangles represent off-chain databases, and ellipses represent off-chain servers.

contract holds data for each position, including collateral quantity and position volume, making it a crucial part of the risk control process (Chen et al., 2024).

2.4.2 Stakeholders

In Oracle Pricing Perpetual Futures exchanges, there are three primary stakeholders: traders, liquidity providers, and the protocol. Each group has distinct roles and objectives within the ecosystem:

1. Traders: are the primary participants in the exchange, and they can be categorized into two main types, speculators and arbitrageurs. Speculators aim to prove their market belief is correct by taking positions that reflect their expectations of future price movements. They seek to profit from these predictions, taking on risk in the hope of earning future returns. Arbitrageurs engage in risk-free strategies that capitalize on imbalances across different platforms. They seek to profit from these discrepancies without exposure to significant market risk. By arbitraging fees (like funding rate and price impact), they help to ensure price consistency and efficiency across markets, contributing to the overall stability of the financial ecosystem;

2. Liquidity Providers (LPs): play a crucial role by staking their capital into the liquidity pool or vault. Their primary goal is to earn a passive return that compensates for the risk they

undertake. LPs accrue trading fees as compensation for providing liquidity. These fees are proportional to the volume of trading activity and the amount of capital staked in the pool and the distribution implemented by the protocol. The return for LPs is aligned with the level of risk they face, which varies based on market conditions and the performance of the traders. When traders have profitable trades, resulting in a positive PnL (Profit and Loss), there is an outflow of capital from the liquidity pool to the traders. Conversely, when traders incur losses, resulting in a negative PnL, the liquidity pool benefits from an inflow of capital. Thus, the profitability of traders directly impacts the returns for liquidity providers;

3. Protocol: The protocol consists of smart contracts that coordinate the underlying infrastructure, including oracles and keepers, and facilitate trading activities on the exchange. It provides the technological framework for asynchronous, on-chain trade execution, ensuring that trades are conducted efficiently and securely while maintaining the integrity of the exchange. The protocol collects fees from trading activities to cover infrastructure costs and generate revenue. This revenue supports ongoing development and innovation, ensuring the continuous improvement and sustainability of the platform.

The synergy between traders, liquidity providers, and protocol creates a dynamic ecosystem where each stakeholder plays a vital role. Traders drive market activity, liquidity providers ensure the availability of capital for the asynchronous trading, and the protocol provides the necessary infrastructure and continuous improvements to facilitate seamless trading experiences. To illustrate these interaction, Figure A.6 in Appendix A represents the capital flow between the mentioned stakeholders in Ostium Labs (2024b) platform.

2.4.3 Fees

Fee structures and rates vary significantly across exchanges. Chapter 3 analyzes the fee structures implemented in major Oracle Pricing Perpetual Futures DEXs and proposes standardized category names to label similar fees. For now, let's examine the most common fees without going into extensive detail.

The most common fees charged are as follows: To open a position, a trader must pay for the gas fees (a cost specific to the blockchain) and execution costs. Since there is no internal price discovery, the protocol applies a virtual spread, simulating the market dynamics of the underlying asset. These fees are also charged again when the position is closed. While the position is open, fees such as the funding rate and borrowing fee are charged per block, which is a specific way blockchain perceives time.

Oracle pricing perpetual futures exchanges are distinct in their asynchronous, on-chain trade execution, providing virtual price exposure without the need for an immediate counterparty. This model diverges from the one used in CEXs, employing a vault or liquidity pool funded by LPs to settle trades. Fee structures in this model are critical as they price the risk of positions to the protocol, balancing open interest (OI) skew (difference between long and short opened positions) and mitigating counterparty risk.

The liquidity pool acts as a third counterparty, akin to AMMs, but instead of facing impermanent loss, LPs encounter directional exposure, directly offsetting trader profits and losses. Effective fee structures are vital to minimize directional exposure for LPs, thereby reducing the variability in their returns.

2.4.4 Risks and Mitigation Strategies

Oracle pricing DEX introduces significant risk factors that must be meticulously managed. Oracles are external data providers that feed real-time price information into the blockchain, ensuring that the prices reflected in smart contracts are accurate and up-to-date. However, the decentralized nature of oracles makes them susceptible to manipulation and inaccuracies, which can lead to significant discrepancies in asset prices. These discrepancies can be exploited by malicious actors to execute arbitrage attacks or manipulate market prices, resulting in substantial financial losses for traders and liquidity providers. Therefore, robust risk management strategies, including multiple data sources and sophisticated validation mechanisms, are essential to safeguard the integrity of oracle pricing in DEXs (Chaos Labs, 2024).

Chaos Labs is at the forefront of risk management in decentralized finance, specializing in the development of advanced simulation and analysis tools to enhance the security and stability of DeFi protocols. Leveraging sophisticated Agent-Based Simulations (ABS) and other cutting-edge methodologies, Chaos Labs conducts comprehensive stress tests and economic evaluations to identify potential vulnerabilities and optimize risk parameters. By collaborating closely with protocols like Ostium, Chaos Labs aims to create resilient trading environments that balance capital efficiency with robust risk mitigation, ensuring that DeFi platforms can thrive in the volatile and rapidly evolving crypto markets (Chaos Labs, 2024).

Decentralized exchanges that rely on the oracle pricing model face critical risks that can impact their stability and security. These include:

1. Counterparty Risk: refers to the possibility that the other party in a financial transaction may default on their obligations. This risk is particularly relevant in decentralized finance (DeFi) where transactions are conducted without traditional intermediaries. One significant cause of counterparty risk is directional exposure. Directional exposure arises when the market becomes heavily skewed in one direction (either long or short), leading to potential loss and significant variability in LPs (Liquidity Providers) rewards. This skew can increase the likelihood of default as counterparties may face significant losses if the market moves against their positions (Goldberg et al., 2023b);

2. Price Manipulation: involves deliberately influencing the market price of an asset to create artificial price movements. This typically involves placing large buy or sell orders to inflate or deflate prices, thereby creating an opportunity to profit from the manipulated price changes. Such activities can lead to significant volatility and disrupt the natural price discovery process, ultimately undermining market integrity and fairness (Goldberg et al., 2023b);

3. Economic Risk: in oracle pricing models, economic risk is associated with the volatility and liquidity dynamics of the underlying assets. These risks can lead to significant financial losses for traders and liquidity providers if not properly managed (Goldberg et al., 2023a);

Directional exposure is a crucial element in the stability and risk management of oracle pricing perpetual futures exchanges. It refers to the imbalance in open interest towards predominantly long or short positions. This exposure is particularly significant as it directly impacts the rewards for LPs and the overall financial health of the exchange. An unbalanced open interest skew can lead to substantial payouts to traders if the market moves in their favor, potentially depleting the liquidity pool. Conversely, if the market moves against the skew, the liquidity pool may benefit. However, this unpredictability introduces risk, making effective management of directional exposure essential for maintaining equilibrium and ensuring the sustainability of the exchange.

For instance, consider the following scenarios where the liquidity pool is not accruing fees to gain a better understanding of the underlying dynamics of an oracle pricing model:

Scenario 1: Trader vs LP Trader 1 opens a BTC long \$1000 position; BTC price increases 1%; Trader 1 profits \$10; Liquidity pool loses \$10.

Scenario 2: Trader vs Trader Trader 1 opens a BTC long \$1000 position; Trader 2 opens a BTC short \$1000 position; BTC price increases 1%; Trader 1 profits \$10; Trader 2 loses \$10; Liquidity pool balance does not change.

Scenario 3: Trader vs Trader-LP Trader 1 opens a BTC long \$1000 position; Trader 2 opens a BTC short \$500 position; BTC price increases 1%; Trader 1 profits \$10; Trader 2 loses \$5; Liquidity pool loses \$5.

In Scenario 1, there is a long OI skew (difference between long and short OI) of \$1000, and a price movement of 1% creates an outflow from the liquidity pool. However, if the BTC price decreased, the liquidity pool would increase by about \$10. In Scenario 2, the liquidity pool balance remains the same regardless of price movements because the open interest is balanced (1000-1000=0). Scenario 3 falls between Scenario 1 and Scenario 2, where Trader 1 profits \$10 and the liquidity pool loses \$5, sharing the loss with Trader 2, who loses \$5.

In the context of Oracle Pricing Models, several strategies can be employed to mitigate counterparty risk, ensuring a stable and sustainable trading environment. Ostium Labs (2024a) mentions the key characteristics protocols use to mitigate risks:

- Listed Assets: including uncorrelated assets such as commodities or Forex in the exchange's asset listings can significantly reduce counterparty risk. By diversifying the range of assets available for trading, the exchange minimizes the impact of volatility in any single asset class. This diversification ensures that extreme market movements in one asset do not disproportionately affect the liquidity pool, thereby maintaining a more balanced risk profile Ostium Labs (2024a);

- Trading Parameters: setting the right trading parameters is crucial in controlling exposure and managing risk. These parameters include leverage limits to prevent traders from taking excessively large positions, clear liquidation ratios to ensure positions are closed before incurring losses bigger than the collateral, and open interest limits to prevent large imbalances that could lead to directional exposure risks;

- Fee Structure and Parameters: implementing a dynamic and risk-sensitive fee structure can effectively balance risk by incentivizing or disincentivizing certain trading behaviors. This includes charging higher fees for trades that increase open interest skew to discourage imbalances and adjusting fees based on market volatility to compensate the liquidity pool for increased risk during turbulent periods;

- Pool Structure and Incentives: Using asset-specific pools rather than a unified pool approach can help diversify risk and reduce concentration. This involves dividing liquidity into separate pools for different assets to contain risk within each pool and offering incentives for liquidity providers, such as higher returns or reduced fees, to maintain a healthy level of Total Value Locked (TVL) and ensure sufficient reserves to cover potential payouts.

By implementing these risk mitigation strategies, oracle pricing perpetual futures exchanges can better manage counterparty risk, ensuring a more stable and predictable market environment for both traders and liquidity providers. To validate the effectiveness of these risk management strategies, protocols must continuously monitor variables that are closely associated with these risks.

2.4.5 Evaluation Metrics

Monitoring and managing the various risks associated with Oracle Pricing Models is critical. A well-designed dashboard serves as an essential tool for stakeholders, offering real-time insights and historical data to inform strategic decisions, ensure stability, and optimize performance. By analyzing dashboards from Chaos Labs and leading platforms such as GMX and gTrade, the primary metrics include:

1. Volume: total volume measures the cumulative trading activity on the platform, indicating overall liquidity and user engagement. Daily volume tracks trading activity on a day-to-day basis, providing granular insights into market behavior. Volume by asset breaks down trading activity by individual assets, highlighting the most actively traded pairs;

2. Fees: total fees represent the cumulative revenue generated from transaction fees, reflecting the platform's profitability. Cumulative fees track the accumulation of fees over time, indicating long-term revenue growth. Fee revenue breakdown by order execution details the revenue generated from different types of orders, offering insights into the most profitable trading activities. Rolling and price impact fees analyze the costs associated with rolling positions and the impact of trading on fees, helping to understand the fee structure and user costs;

3. Open Interest: long and short open interest measures the total value of outstanding positions, indicating market sentiment and risk exposure. OI imbalance tracks the difference between long and short positions, highlighting potential market pressures and risks;

4. Trades: number of trades counts the total executed trades, indicating trading activity and user engagement. Average collateral and leverage provide insights into risk-taking behavior. Liquidations track the number of forced position closures due to insufficient collateral, indicating market volatility and risk management effectiveness. PnL measures the profit and loss of traders measuring traders' profitability. Net PnL before fees provides raw trading performance data without the influence of fee structures;

5. Users: total users indicate the overall user base, reflecting the platform's popularity and reach. New users track the influx of new participants, providing insights into growth and user acquisition effectiveness;

6. Vault or Liquidity Pool: Total Value Locked (TVL) measures the total value of assets secured in the protocol, indicating user trust and platform security. Staking APR (Annual Percentage Rate) represents the annual return rate for staking, highlighting the attractiveness of staking opportunities. Collateral Ratio measures the ratio of collateral to loans, reflecting the security level and risk management of the vault;

7. Risk: What-if Analysis, by Chaos Labs (2024), assesses potential vulnerabilities based on hypothetical scenarios, aiding in strategic planning and risk management. Alerts provide timely notifications for significant events and risks, ensuring proactive responses. Liquidations monitor the frequency and impact of forced position closures, offering insights into market stability and risk management effectiveness.

2.5 Gap in Background Literature

Despite the increasing importance of oracle pricing models in DeFi, there is a notable scarcity of academic literature dedicated to this specific area. Most of the available information comes from reports produced by Chaos Labs for platforms such as GMX (Goldberg et al., 2023a) and Avantis (Goldberg et al., 2023b). While Chen et al. (2024) provides a useful overview of the perpetual futures landscape, it doesn't go into much detail about ways to mitigate risk and the role of specific fee structure on this type of exchange model. The unique risks that oracle pricing poses for LPs necessitate closer scrutiny.

Each protocol develops its own unique fee structure. While there are similarities across protocols, some fee structures are fundamentally different. For example, gTrade's borrowing fee, known as the borrowing fee, is influenced by different environmental variables than GMX's borrowing fee. These two fees have distinct structures, making direct comparisons challenging. The differences go beyond merely the parameter values.

Currently, there is a lack of articles that offer a comparative analysis or propose methodologies to evaluate the performance of different fee structures in Oracle Pricing Models. This gap highlights the need for comprehensive research to develop standardized methods for assessing and optimizing fee structures to ensure the sustainability and efficiency of these models.

Chapter 3

Fee Structure Categorization

Oracle Pricing Models fees are crucial for the overall stability and functionality of the platform. Each protocol designs its own unique fee structure to meet specific needs and objectives. While similarities exist across different protocols, fundamental differences make direct comparisons challenging. These distinct structures underscore the complexity of fee design and the importance of understanding nuanced differences beyond mere parameter values.

This chapter proposes a method to categorize fees based on their structure, their influence on trading components and the timing of their charge. It includes a comprehensive comparison of the fees used by GMX, gTrade, and Avantis, providing a clear understanding of the variables that these fees depend on.

3.1 Methodology

The methodology to categorize fees was developed to provide a clear and systematic understanding of the diverse fee structures across various protocols. The following steps outline the process:

1. Select Protocols with Fundamentally Different Fee Structures: identify a range of protocols that employ distinct approaches to fee structuring. This ensures a comprehensive analysis that captures the variety in how fees are implemented. The protocols chosen for this analysis include GMX, gTrade, and Avantis, each known for their unique fee models;

2. Identify All Fees Charged by Each Protocol: conduct a thorough examination of each protocol to list all the fees they charge. Collect detailed documentation and resources from each protocol;

3. Detailed Description of Each Fee: for each identified fee, gather a detailed description to understand the variables it depends on. This step focuses on the structure of the fee rather than the specific values charged, which would be more relevant for a competitive analysis. Analyze how each fee is calculated, including any dependencies on factors like position size, OI skew, volatility, and utilization rates;

4. Identify Similarities in Target Position Element and Timing: compare the fees across protocols to identify similarities based on the target position element they affect—such as price, collateral, or PnL. Categorize fees based on the timing of their application, such as whether they are charged at the opening/closing of a position or during the holding period;

5. Propose Fee Categories: Develop a set of fee categories that encompass the identified fees, providing a framework to understand and compare the different fee structures. Ensure that these categories are comprehensive enough to accommodate the majority of fees.

The process followed the above order, but for better comprehension, the new nomenclature is presented first, followed by an in-depth analysis of each fee structure.

3.2 Fee Categorization Nomenclature

To enhance clarity and standardization, a new nomenclature for categorizing fees has been developed. This categorization is based on two primary dimensions: the target position element affected by the fee and the timing of the fee application.

Through a comprehensive examination of fees from GMX, gTrade, and Avantis, the following categories were established to address different components of the position at specific times. Each fee structure analyzed fell within these defined categories. In the subsequent sections, each category is described in more detail, along with a breakdown of fees by each protocol within their respective fee category.

The different steps in the methodology were followed, and these are the proposed categories:

1. Order Execution: charged on the collateral at the time of order execution, whether opening or closing a position. These fees cover various infrastructure costs associated with running the trading platform, including blockchain operations, oracle services, and keeper functions. Additionally, a portion of these fees contributes to LPs revenue and supports comprehensive risk management strategies. The fee structure is designed to ensure the smooth operation of the trading platform while compensating LPs and managing systemic risks;

2. Rolling: time-based charges, that accumulate per block, follow a specific calculation logic and target the position PnL. These fees can be structured in a way that they are neutral to the trading platform (zero-sum), meaning that they might be redistributed among traders based on certain conditions, or they can be allocated to a vault. This fee type helps manage the financial dynamics of positions held over time, aligning the costs with the duration for which the positions are maintained;

3. Price Impact: adjusts the execution price relative to the market price at the time of the transaction. This fee reflects the real-world trading constraints of the underlying market, such as the spread between bid and ask prices, slippage due to order size, and protocol skewness. Price Impact aims to simulate the actual market conditions traders would face,

providing a more realistic trading environment and helping to manage the market impact of large or sudden trades. This fee category is crucial for maintaining market stability and ensuring that the price reflects both market demand and protocol-specific characteristics.

Table 3.1 categorizes fees in oracle pricing models by their target and timing. The order execution fee targets collateral and is applied when positions are opened or closed, reflecting the cost of entering and exiting the market. The rolling fee targets PnL during the holding period, accounting for the ongoing cost of maintaining a position. Lastly, the price impact fee targets the price and is also applied during the opening and closing of positions, capturing the effect of trade execution on market prices.

Table 3.1: Categorization of fees in oracle pricing models

Category	**Target**	**Timing**
Order Execution	Collateral	Open/Close
Rolling	PnL	Holding
Price Impact	Price	Open/Close

3.3 Order Execution

Here the order execution fees across different trading protocols are compared, highlighting their characteristics. Table 3.2 provides a clear comparison of these fees regarding timing and if they are dependent on the skew.

Table 3.2: Order execution fees across different protocols

Protocol	**Name**	**Timing**	**Skew**
GMX	Opening Fee	Open	X
	Closing Fee	Close	X
gTrade	Opening Fee	Open	
	Closing Fee	Close	
Avantis	Dynamic Opening Fee	Open	X
	Closing Fee	Close	

GMX charges fees that are a percentage of the total position size for opening and closing trades. These fees follow the Market/Taker dynamics (if the trader decreases the OI skew, he is a maker, otherwise, he is a taker) which can only assume two possible values. The percentage depends on the OI skew, allowing GMX to adjust the trading cost in response to OI skew, thereby incentivizing traders under different market scenarios.

When traders open and close positions on the gTrade platform, they are charged a fee that is a percentage of the total position size. This fee percentage remains constant and does not vary with market conditions or skew, providing a predictable cost structure for traders upon entry.

Avantis charges a dynamic opening fee that is not only a percentage of the total position size but also varies linearly depending on the OI skew. This flexible fee structure is designed to balance trading activities and mitigate systemic skew, adjusting the cost of trade entry to better align with current market dynamics and protocol trading risks. On the other hand, the closing fee on Avantis, while still a percentage of the total position size, remains constant and does not depend on the OI skew. This approach provides stability and predictability for traders looking to close positions, ensuring that the exit costs are consistent regardless of the prevailing market conditions.

3.4 Rolling

This section analyzes the rolling fees across various trading protocols, focusing on how these fees are influenced by factors such as skew, utilization, volatility, and the inclusion of other assets. Table 3.3 illustrates the differences between protocol-specific fees.

Table 3.3: Rolling fees across different protocols

Protocol	**Name**	**Skew**	**Utilization**	**Volatility**	**Other Assets**
GMX	Funding Rate	X			
	Borrowing Fee	X	X		
gTrade	Borrowing Fee	X	X	X	X
Avantis	Dynamic Margin Fee	X	X		

The GMX protocol utilizes a velocity-based Funding Rate fee, which is primarily influenced by the OI skew. This fee reflects the directional bias of the market and helps to balance long and short positions by periodically adjusting the cost paid or received by traders based on their open positions. In addition to the Funding Rate, GMX charges a Borrowing Fee that depends on both the OI skew and the pool utilization rate. This fee compensates for the risk and cost associated with providing liquidity in leveraged positions and adjusts according to the demand for borrowing (utilization) and the direction of the market trades (skew).

The gTrade Borrowing Fee is influenced by multiple factors including OI skew, utilization, volatility, and other asset characteristics. This comprehensive approach allows the protocol to dynamically adjust fees based on current market conditions, the frequency and volume of asset borrowing, and inherent market risks associated with volatility, and specific asset characteristics like correlation.

Avantis implements a Dynamic Margin Fee that adjusts based on OI skew and utilization. This fee is designed to reflect the real-time risk associated with varying degrees of leverage used by traders and the overall demand for margin within the system.

3.5 Price Impact

This section examines the price impact fees across various trading protocols, emphasizing how these fees are influenced by factors such as order size and market skew. Table 3.4 presents a comparison of these spread fees across different protocols.

Table 3.4: Price impact fees across different protocols

Protocol	Name	Size	Skew
GMX	Price Impact	X	X
gTrade	Fixed Spread		
	Dynamic Spread	X	X
Avantis	Dynamic Spread	X	X

GMX incorporates a Price Impact Fee influenced by both the total position size and OI skew. This fee structure is designed to account for the impact a trader's position size can have on the market price, particularly in less liquid markets or during volatile trading sessions. The skew of the market—indicating the balance between buy and sell orders—also plays a crucial role in determining the fee, as larger trades in a skewed market could significantly move prices, thereby warranting a higher fee to mitigate potential market disruption.

gTrade utilizes both Fixed and Dynamic Spreads. The Fixed Spread remains constant, independent of the trade's size or the OI skew. Alongside the fixed component, gTrade also applies a Dynamic Spread that adjusts based on the total position size and the OI skew. This additional spread is tailored to reflect the real-time liquidity conditions and the prevailing balance between long and short.

Avantis also adopts a Dynamic Spread, which adjusts according to position size and the prevailing OI skew. This method allows the platform to tailor trading costs in real-time, aligning fees with the risk associated with the current protocol state and the directional push of trades. This fee ensures that traders contributing to larger price movements or trading in skewed markets compensate for the increased risk and potential market impact they introduce.

3.6 Looking for the optimal fee structure

As described in this chapter, different protocols often use the same terminology to refer to different types of fees, highlighting a significant standardization problem across exchanges. Beyond this inconsistency, another critical question arises: what constitutes the optimal fee structure?

Logically, if a fee structure is designed to depend on the characteristics of the exchange—charging more whenever the directional exposure for LPs increases—it should ideally help reduce the skewness by increasing charges for positions that amplify imbalances. However, if these adjustments are not carefully calibrated, they may not effectively address the skewness problem, potentially leading to increased systemic risk instead of mitigating it.

Currently, the literature lacks a comprehensive identification and comparison of the different fee structures used in oracle pricing exchange models. Moreover, there is no established methodology for analyzing and comparing the performance of these fee structures. To address this gap, the next chapter introduces a robust methodology designed to analyze and measure the performance of various fee structures, providing a systematic framework for determining the most effective approach.

Chapter 4

Methodology for Evaluating Fees

This chapter proposes a way to systematically analyze and compare different fees by designing a quantitative model to simulate the trading engines used by oracle pricing models. This simulation incorporates the modular fee structures identified in the previous chapter. Using agent-based simulation, traders are represented as agents to accurately mimic real-world trading behaviors. Additionally, Monte Carlo simulations will be employed to compute multiple price paths using Geometric Brownian Motion, generating a large number of samples and increasing precision in estimating evaluation parameters. By replicating the intricate interactions and price fluctuations within the trading environment, this quantitative model aims to provide a robust framework for evaluating the efficiency and effectiveness of various fees.

4.1 Agent-Based Simulation

Agent-Based Simulation (ABS) has emerged as a powerful tool for modeling complex systems, particularly in the context of financial markets. This section provides an overview of the fundamentals of ABS, its relevance in oracle pricing models, and the specific implementation approach for this study.

4.1.1 Fundamentals of Agent-Based Simulation

Agent-Based Simulation (ABS) involves the modeling of systems as a collection of autonomous, interacting agents. Each agent operates based on a set of rules and can represent various entities such as individuals, organizations, or even biological cells. The fundamental strength of ABS lies in its ability to capture emergent phenomena, which arise from the interactions of individual agents following simple behavioral rules. These emergent phenomena are often non-linear and cannot be easily predicted by analyzing the components of the system in isolation (Wang, 2014).

In the context of financial markets, agents typically represent market participants, such as individual investors, banks, and trading firms. Each agent makes decisions based on its own objectives, information, and strategies. The collective behavior of these agents can generate complex market dynamics, including price formation, trading volume fluctuations, and the propagation of shocks through the financial system (Yu et al., 2015).

In the context of oracle pricing models, ABS is particularly valuable for analyzing the optimal parameters for protocol fees. These simulations allow for the modeling of diverse market participants, such as speculators and arbitrageurs, and their interactions within the trading environment. For instance, Goldberg et al. (2023a) has utilized ABS to simulate and understand the dynamics within DeFi protocols, demonstrating the utility of this approach in managing protocol risk and optimizing fee structures.

4.1.2 Implementation of Agent-Based Simulation in This Study

To systematically analyze and compare different protocol structures, this study will implement an ABS model tailored to oracle pricing models. The implementation will involve the following steps:

1. Defining Agents: the simulation will include various types of agents such as speculators, who are not influenced by fees, and arbitrageurs, who open trades based on open interest imbalance thresholds. Traders will be modeled with unlimited resources, constrained only by opening rates and holding periods;

2. Environment Setup: a virtual market environment will be created to mimic real-world conditions of perpetual futures markets. This includes setting up the rules for market operations, fee structures, and mechanisms for price discovery, which is external, and trade execution. Key assumptions include:

    - Order execution fees impact only the profit and loss (PnL), not the collateral;
    - Open interest will be calculated using the collateral currency rather than the number of contracts;
    - All fees will accrue to the vault and are not shared with the protocol;
    - Only one asset is traded.

3. Simulation Execution: the Monte Carlo simulations will be employed to compute multiple price paths using Geometric Brownian Motion (GBM), generating a large number of samples to increase precision in estimating evaluation parameters.

4. Data Analysis: data generated from the simulations will be collected and analyzed to understand the impact on PnL. This analysis aims to identify the optimal fee structures that balance the needs of different market participants while ensuring the robustness of the pricing mechanism.

By implementing ABS in this study, it replicates the intricate interactions and price fluctuations within the trading environment, providing a robust framework for evaluating the efficiency and effectiveness of various fees in Oracle Pricing Models.

4.2 Simulation Model Architecture

The trading model architecture for simulating Oracle Pricing Models requires several key components to operate effectively, following the steps in Algorithm 1.

input : A list *Price*; Parameters for *Trader* and *Fee*
output: Dataframe *Positions*
iterate N price paths;
for $n \leftarrow 1$ **to** N **do**
  Model $\leftarrow$ `TradingEngine` (*Price*[*n*], *Trader*, *Fee*);
  iterate T periods;
  for $t \leftarrow 1$ **to** T **do**
    `Step` (Model);
  end
  Positions[*n*] $\leftarrow$ `Output` (Model);
end

Algorithm 1: Simulation model steps

These components include price, trader, and fee. The following sections elaborate on each component and their roles within the trading engine.

1. Price: unlike order-book models, where prices are discovered internally through bid and ask orders, the oracle pricing model uses external price inputs. These prices serve as the basis for executing trades and calculating profits and losses (section 4.3);

2. Trader: the model includes logic for opening and closing positions. This logic simulates the decision-making processes of traders, encompassing speculators who are insensitive to fee values and arbitrageurs who react to open interest imbalances (section 4.4);

3. Fee: this component defines the structure and amount charged for trading activities. Fees can vary based on different parameters and are essential for evaluating their impact on protocol performance (section 4.6).

These inputs form the foundation of the trading engine (section 4.5), which runs multiple simulations using Monte Carlo methods (Harrison, 2010). Each simulation iteration generates a set of positions, representing all trades opened and closed during the simulation. The outcomes of these positions allow for the analysis of trader profitability, charged fees, and open interest imbalances (section 4.7).

Figure 4.1 illustrates the simulation model architecture.

4.3 Price

Brownian motion, named after the botanist Robert Brown, describes the random movement of particles suspended in a fluid, resulting from their collision with fast-moving molecules in the fluid. In the context of finance, Brownian motion has been adapted to model the random behavior

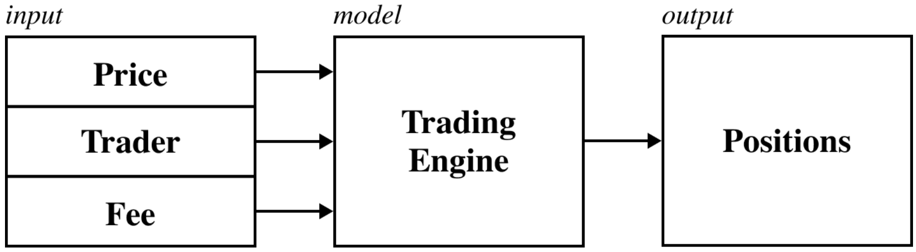


Figure 4.1: Oracle Pricing Model simulation architecture

of asset prices over time. The Geometric Brownian Motion is widely used to simulate and predict the future price paths of financial assets (Ibrahim et al., 2021).

GBM is a stochastic process used in this model, characterized by several key assumptions. Firstly, it assumes that asset prices follow a log-normal distribution, which means that the logarithm of asset prices follows a normal distribution. Additionally, the increments of the asset price are considered independent of each other, ensuring that past price movements do not influence future movements. Lastly, GBM assumes constant drift and volatility over time, meaning that the mean rate of return (drift) and the standard deviation of returns (volatility) remain unchanged throughout the simulation period. These assumptions provide a foundational framework for simulating asset price dynamics in a trading environment (Ibrahim et al., 2021).

GBM is defined by the stochastic differential equation: Geometric Brownian Motion is defined by the stochastic differential equation:

$$dS_t = \mu S_t\,dt + \sigma S_t\,dW_t$$

In this model, S_t represents the asset price at time t. The drift coefficient, denoted as μ, represents the average return. The volatility coefficient, σ, represents the standard deviation of returns. Additionally, dW_t is a Wiener process or standard Brownian motion, which introduces randomness into the asset price movements (Ibrahim et al., 2021).

The assumption of independence is a limitation, as real asset prices often exhibit correlations over time, which GBM does not capture. Another limitation is the assumption of constant volatility, whereas volatility in financial markets typically changes over time. Additionally, GBM does not account for long-term memory, meaning it does not consider long-range dependencies in asset prices, which are observed in real market data (Ibrahim et al., 2021).

4.3.1 Application in Simulation

In simulations, GBM is used to model the future price paths of assets within an agent-based framework. This involves:

- Initial Parameters: setting the initial price, drift, and volatility parameters based on historical data.

- Stochastic Process: using the GBM model to generate random price paths. At each time step, the price is updated according to the GBM equation.
- Multiple Simulations: Running multiple simulations to create a distribution of possible future prices, helps in risk assessment and decision-making by providing an extensive sample of prices.

The Geometric Brownian Motion model is valuable in agent-based simulations as it provides a straightforward yet powerful method to represent the random nature of asset prices. It is particularly well-suited for oracle pricing model simulations, which require generating multiple price paths from the same distribution to accurately reflect potential market scenarios.

4.4 Trader

In the simulation model, traders are central to the dynamics of the market, making various actions and decisions that influence market outcomes. Traders are categorized into two main groups: speculators and arbitrageurs, each with distinct goals and strategies.

Traders in the simulation decide on the following variables:

- Opening Period: the period when the trade is opened;
- Leverage: the amount of leverage selected for the position;
- Collateral: the specific amount of collateral used to fund the position;
- Direction: the decision to take a long (buy) or short (sell) position;
- Closing Period: the period at which the trader closes the position.

To provide greater granularity to trading decisions, we classify traders into two high-level groups:

1. Speculators: these traders bet on the price direction of an asset with the primary goal of profiting from market movements. Speculators can employ various strategies, such as moving average crossover, mean reversion, momentum trading, and random trades;
2. Arbitrageurs: these traders aim to profit from imbalances in open interest (OI). They engage in arbitrage activities between spot markets and perpetual futures markets. Their strategies often include taking advantage of funding rates and execution price discrepancies.

4.4.1 Speculator

Speculators open positions at an average rate of λ_{open} and hold them for an average duration of $1/\lambda_{hold}$ periods. The number of trades opened by speculators at time t follows a Poisson distribution (eq. 4.1), while the holding period for each position follows an Exponential distribution (eq. 4.2).

$$opening\_positions_t \sim Poisson(\lambda_{open}) \tag{4.1}$$

$$hold\_periods_i \sim Exp(1/\lambda_{hold}) \tag{4.2}$$

The number of trades opened by speculators at time t is given by $opening\_positions_t$. The holding period $hold\_periods_i$ represents the number of periods a position is kept open. Therefore, the closing period is calculated as: $t_i^{close} = t_i^{open} + hold\_periods_i$.

The leverage of position i follows a Normal distribution (eq. 4.3) with mean μ_{lev} and variance σ_{lev}^2, represented by the variable L_i. The collateral of position i follows a Lognormal distribution (eq. 4.4) with mean μ_{col} and variance σ_{col}^2, represented by the variable C_i.

$$L_i \sim Normal(\mu_{lev}, \sigma_{lev}^2) \tag{4.3}$$

$$C_i \sim Lognormal(\mu_{col}, \sigma_{col}^2) \tag{4.4}$$

The distribution of the variable C_i was determined through an analysis performed on the dataset from Dune Analytics gTrade Stats. Using the Kolmogorov-Smirnov test with a 95% confidence interval, we tested the data for normality. The resulting p-value was approximately 0, leading us to reject the null hypothesis that the sample data follows a normal distribution. Subsequently, we tested if the data fit a lognormal distribution. In this case, we do not reject the null hypothesis, as the p-value was 0.68.

A similar analysis was conducted on the leverage sample data. However, neither the normal nor the lognormal distribution provided a good fit. For simplicity and practicality in modeling, we assume a normal distribution for leverage.

The decision between opening a long or short position is dependent on the following strategies used by traders:

1. Moving Average Crossover Strategy: the algorithm calculates the short-term and long-term moving averages and generates a signal based on their crossover (Algorithm 2 in Appendix B);

2. Mean Reversion Strategy: the algorithm calculates the mean and standard deviation of recent prices and generates a signal if the current price deviates significantly from the mean (Algorithm 3 in Appendix B);

3. Momentum Strategy: The algorithm calculates the rate of change (ROC) over a specified window and generates a signal based on whether the ROC exceeds certain thresholds (Algorithm 4 in Appendix B);

4. Random Strategy: The algorithm assigns a probability to open a long position and uses the complementary probability to open a short position (Algorithm 5 in Appendix B).

4.4.2 Arbitrageur

Arbitrageurs can employ various strategies to profit from market inefficiencies. Among these, funding rate arbitrage involves taking advantage of differences in funding rates between perpetual futures contracts and the spot market. However, the methodology presented here focuses on price impact arbitrage, where traders exploit price discrepancies caused by large trades impacting the market price. This strategy is highlighted due to its relevance in assessing the immediate effects of trading activities on market dynamics.

The first Algorithm 6 in Appendix B, step to open arbitrageur position, determines whether a new position should be opened based on the open interest imbalance, denoted as $\Delta OI(t)$, and a predefined threshold. The algorithm receives these two inputs and outputs a Boolean value indicating whether to open a position. If the OI imbalance meets or exceeds the threshold, the algorithm returns true, signaling that a position should be opened. Otherwise, it returns false, indicating no position should be initiated. This logic ensures that new positions are only considered when significant market imbalances occur, aligning trader actions with prevailing market conditions.

The second Algorithm 7 in Appendix B, get arbitrageur position details, sets the specific details of a new trading position based on the OI imbalance and an OI cap. The algorithm first checks the sign of $\Delta OI(t)$ to determine the position's direction: if $\Delta OI(t)$ is positive, a short position is set; if negative, a long position is set. The leverage is fixed at 1, indicating no additional borrowing beyond the initial collateral. The collateral required for the position is then calculated as the absolute value of $\Delta OI(t)$ multiplied by the OI cap.

The third Algorithm 8 in Appendix B, step to close position, decides whether an existing position should be closed based on the current OI imbalance, the direction of the initial position (D_i), and a closing threshold. The algorithm first checks if D_i is 1 (indicating a long position) and then assesses if $\Delta OI(t)$ exceeds the threshold. If both conditions are met, it returns true, signaling that the position should be closed. Conversely, if D_i is -1 (indicating a short position) and $\Delta OI(t)$ is less than the negative threshold, the position is also closed. If neither condition is met, the algorithm returns false, and the position remains open. This logic ensures that positions are closed in response to dynamic fee charges, ensuring the arbitrageurs profits from the trade.

4.5 Trading Engine

This section presents the formulas and variables utilized in the trading engine, which are commonly employed by various protocols to represent trading positions. It is important to note that these formulas are general in nature and do not correspond to any specific protocol.

Table 4.1 illustrates the core variables in the trading engine. The set $\mathscr{I}$ represents the set of all positions denoted by i, including all possible positions in the analysis. $\mathscr{T}_i$ denotes the set of time intervals, t, during which position i is open, specifying the active periods for each position. L_i indicates the leverage for position i, which is the ratio of the position's value to the equity used. C_i represents the collateral required or used for maintaining position i. D_i indicates the direction of position i, with 1 for a long position and -1 for a short position, showing whether the position is betting on the price going up or down. $P(t)$ is the price at time t, representing the asset's price at a specific time. t_i^{open} is the opening time for position i, the specific time when the position was opened, and t_i^{close} is the closing time, when the position was closed. τ_i^{open} is the order execution fee rate at opening for position i, indicating the fee rate charged when opening the position, and τ_i^{close} is the order execution fee rate at closing. $\phi_i(t)$ is the rolling fee rate at time t for position i, the fee rate for the fee charged every period. ζ^{open} measures the price impact fee rate at opening, and ζ^{close} measures the price impact fee rate at closing, both indicating how much they affect execution price. Finally, OI_{cap} is the maximum open interest capacity, representing the maximum allowable open interest or the total number of opened positions size.

Table 4.1: Trading Engine Variables

Variable	**Description**
$\mathscr{I}$	Set of all i positions
$\mathscr{T}_i$	Set of t time intervals for which position i is open
L_i	Leverage for position i
C_i	Collateral for position i
D_i	Direction of position i (1 for long, -1 for short)
$P(t)$	Price at time t
t_i^{open}	Opening time for position i
t_i^{close}	Closing time for position i
τ_i^{open}	Order execution fee rate at opening for position i
τ_i^{close}	Order execution fee rate at closing for position i
$\phi_i(t)$	Rolling fee rate at time t for position i
ζ^{open}	Price impact fee rate at opening
ζ^{close}	Price impact fee rate at closing
OI_{cap}	Maximum Open Interest capacity

Table 4.2 represents the main formulas and variables. The variable PnL_i represents the profit and loss of position i. $gPnL_i$ denotes the gross profit and loss of position i, representing the position's PnL before fees. The variable $gPV_i(t)$ represents the gross position value at time t, measuring position size after PnL. EF_i is the order execution fee for position i, accounting for opening and closing. RF_i denotes the rolling fee for position i, accumulated per period. The variable PF_i represents the price impact fee for position i, a short form from substituting p_i^{open}, the execution price at opening, and p_i^{close}, the execution price at closing. $uPnL_i(t)$ represents the unrealized PnL at time t, the same as if the position closed at time t. $OI_i(t)$ denotes the open interest for position i at time t, having a positive value if the position is opened. The variable $OI_{long}(t)$ represents the total open interest for long positions and $OI_{short}(t)$ the total open interest for short positions at time t. $\Delta OI(t)$ represents the net open interest at time t. Finally, $\Delta OI^{norm}(t)$ denotes the normalized net open interest at time t, normalized by the OI_{cap}.

Lastly, this constraint ensures that the trader's PnL never falls below zero. The condition $LR \cdot C_i + uPnL_i(t) > 0$ serves as the liquidation criterion, ensuring that the trader does not incur

Table 4.2: Trading Engine Formulas

Notation	Description	Formula
PnL_i	Profit and Loss of position i	$PnL_i = gPnL_i - EF_i - RF_i - PF_i$
$gPnL_i$	Gross PnL of position i	$gPnL_i = L_i C_i D_i \dfrac{P(t_i^{close}) - P(t_i^{open})}{P(t_i^{open})}$
$gPV_i(t)$	Gross Position Value at time t	$gPV_i(t) = L_i C_i \left(1 + D_i \dfrac{P(t) - P(t_i^{open})}{P(t_i^{open})}\right)$
EF_i	Order Execution Fee for position i	$EF_i = gPV_i(t_i^{open}) \cdot \tau_i^{open} + gPV_i(t_i^{close}) \cdot \tau_i^{close}$
RF_i	Rolling Fee for position i	$RF_i = \sum_{\substack{t \in T_i \\ t \neq t_i^{open}}} gPV_i(t) \cdot \phi_i(t)$
PF_i	Price Impact Fee for position i	$PF_i = L_i C_i \dfrac{P(t_i^{close})}{P(t_i^{open})} \dfrac{\zeta^{open} + \zeta^{close}}{1 + D_i \cdot \zeta^{open}}$
p_i^{open}	Execution price at opening for position i	$p_i^{open} = P(t_i^{open}) \cdot (1 + D_i \cdot \zeta^{open})$
p_i^{close}	Execution price at closing for position i	$p_i^{close} = P(t_i^{close}) \cdot (1 - D_i \cdot \zeta^{close})$
$uPnL_i(t)$	Unrealized PnL at time t	$uPnL_i(t) = PnL_i\big\rvert_{t_i^{close}=t}$
$OI_i(t)$	Open Interest for position i at time t	$OI_i(t) = \begin{cases} L_i C_i & \text{if } t \in \mathscr{T}_i \\ 0 & \text{otherwise} \end{cases}$
$OI_{long}(t)$	Total Open Interest for long positions at time t	$OI_{long}(t) = \sum_{\substack{i=0 \\ D_i=1}}^{\mathscr{I}} OI_i(t)$
$OI_{short}(t)$	Total Open Interest for short positions at time t	$OI_{short}(t) = \sum_{\substack{i=0 \\ D_i=-1}}^{\mathscr{I}} OI_i(t)$
$\Delta OI(t)$	Net Open Interest at time t	$\Delta OI(t) = OI_{long}(t) - OI_{short}(t)$
$\Delta OI^{norm}(t)$	Normalized Net Open Interest at time t	$\Delta OI^{norm}(t) = \dfrac{\Delta OI(t)}{OI_{cap}}$

losses exceeding the collateral. Here, LR represents the liquidation ratio, which ensures that the position is closed before the collateral is completely depleted.

4.6 Fee

In this section, we offer a visual representation of when specific fees are charged, along with a brief explanation of the fee structure definition.

The fees applied to the positions are categorized as following: 1) Order execution (EF); 2) Rolling (RF); and 3) Price Impact (PF). Figure 4.2 illustrates the specific points in time at which these fees are charged on a trading position. The vertical elements in the diagram represent the same period t, while the horizontal elements depict how the vault and trader's unrealized PnL changes concerning position i.

At t_i^{open}, when a request to open a position is made, the vault accrues the opening order execution fee, EF_i^{open}, and the opening price impact fee, PF_i^{open}. In the next period, the rolling fee, RF_i, is charged on the holding position, accumulating the price variation between the previous and current period, $\Delta gPnL_i$. Period $t_i^{open}+2$ corresponds to holding, where the capital flow remains similar to the previous period. Upon closing the order at t_i^{close}, the closing order execution fee, EF_i^{close}, and the closing price impact fee, PF_i^{close}, are charged, along with the rolling fee, RF_i, between t_i^{close} and the previous period. Any price variation, potentially favorable to the trader, is accumulated into the position.

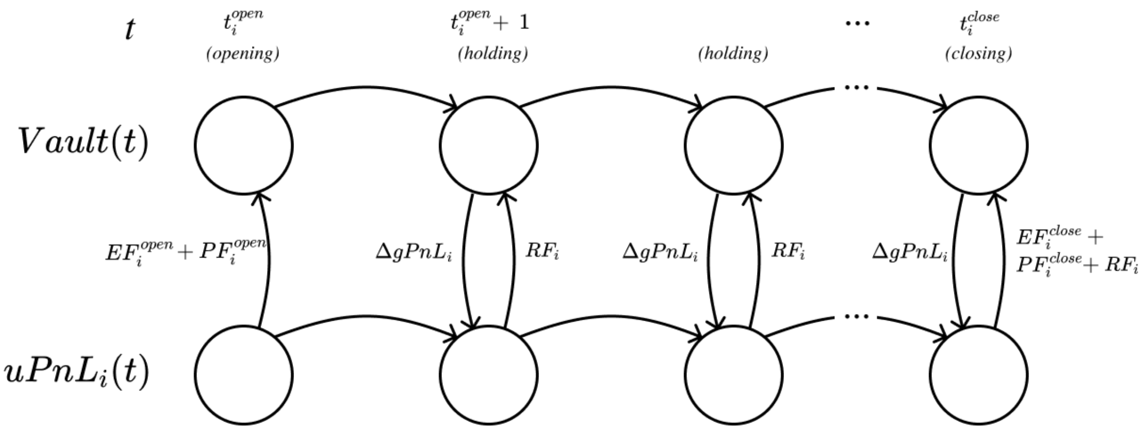


Figure 4.2: Fee Structure Charge-Temporal Representation

Fees can assume different structures, ultimately influencing the overall cost to the trader. In the previous section, the method to compute each fee was introduced. However, the part that introduces the most variability is the definition of each rate for different positions at any point in time. Therefore, it is necessary to specify exactly what the structure and its parameters are. Appendix C provides examples of the most common fee structures with some variations (i.e., fixed or dynamic). Clearly defining these structures is a crucial task when setting up this model.

4.7 Evaluation Criteria

In this section, we define the criteria used to evaluate the effectiveness and impact of different fee structures, as well as the data analysis techniques employed to interpret the simulation results.

The simulation generates N price paths, each with a length of T periods. Running the model across multiple price paths enhances the robustness of the results. To summarize each run, we aggregate the main variables that affect Traders and Liquidity Providers.

Figure 4.3 illustrates the flow of capital between the Vault and Traders. Capital is removed from the vault if the gross PnL ($gPnL$) exceeds the fees collected by the Vault. It is crucial to understand that a positive PnL represents money moving from the vault to the traders. Conversely, a negative PnL indicates capital flowing from traders to the vault. For the sake of maintaining liquidity and ensuring the stability of the protocol, it is expected that the PnL in the long run is negative. This ensures that liquidity providers (LPs) and the protocol are consistently rewarded.

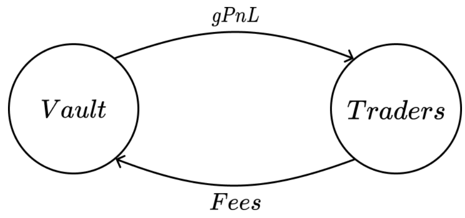


Figure 4.3: Vault and Traders Capital Flow

4.7.1 Value at Risk (VaR)

Value at Risk (VaR) is a statistical technique used to measure the risk of loss for investments. It estimates the maximum potential loss over a specific time period with a given confidence level under normal market conditions. Essentially, VaR answers the question: "What is the most I could expect to lose with a certain level of confidence over a specific period?"

The key components of VaR are:

- Time Horizon: The period over which the risk is assessed (e.g., one day, one month);
- Confidence Level: The probability that the actual loss will not exceed the VaR (common confidence levels are 95%, 99%, and 99.9%);
- Potential Loss Amount: The estimated maximum loss over the specified time horizon at the given confidence level.

Here, VaR is calculated via Monte Carlo Simulation, which uses computer simulations to model a wide range of possible outcomes based on underlying assumptions of the model.

The objective is to assess the downside risk for liquidity providers in Oracle pricing exchange models using VaR. In this model, the cumulative PnL or *sum_PnL* represents the net capital flow between liquidity providers and traders. To assess the VaR, after running the model N times, we compute the percentile corresponding to the considered confidence level.

4.7.2 Summary Statistics

A price path runs from 0 to T periods; however, our objective is to measure the difference between the initial and final states of the Vault and Traders. By summing all positions' PnL from 0 to T, we obtain this value.

Valuable metrics are the **mean** and **standard deviation** of these lists:

- *sum_PnL*: size N
- *sum_Fee*: size N
- ΔOI_{score}: size N

These values are calculated from the dataframe positions, in which Table 4.3 describes the columns in the positions data frame from the simulation output.

Table 4.3: Description of Position Data Frame Columns

Column	Description
n	Price path index
i	Position index for price path n
L	Leverage of position i in price path n
C	Collateral of position i in price path n
D	Direction (long/short) of position i in price path n
t^{open}	Opening period of position i in price path n
t^{close}	Closing period of position i in price path n
$gPnL$	Gross PnL of position i in price path n
EF	Order execution fee of position i in price path n
RF	Rolling fee of position i in price path n
PF	Price impact fee of position i in price path n

The formulas introduced above do not have the n (price path) dimension. The following three formulas have been adapted to consider this dimension while maintaining the same logic.

$$sum\_Fee[n] = \sum_{i \in I_n} EF[n,i] + RF[n,i] + PF[n,i], \forall n \in N \tag{4.5}$$

$$sum\_PnL[n] = \sum_{i \in I_n} (gPnL[n,i]) - sum\_Fee[n], \forall n \in N \tag{4.6}$$

Similar to the Mean Squared Error, this metric measures the square distance between the OI skew and 0:

$$\Delta OI_{score}[n] = \frac{1}{T} \sum_{t \in T} \Delta OI[n,t]^2, \forall n \in N \tag{4.7}$$

4.7.3 Statistical Tests

A 95% significance level is applied to each test to ensure the robustness and reliability of the results.

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (K-S) test is a non-parametric test used to compare a sample with a reference probability distribution or to compare two samples. Using the one-sample K-S test to determine if the sample comes from a normal distribution, the null (H_0) and alternative hypothesis (H_1) are as follows:

- H_0: The sample comes from a normal distribution;
- H_1: The sample does not come from a normal distribution.

Welch's t-test

Welch's t-test is a parametric test used to compare the means of two independent samples, particularly when the variances of the two samples are not assumed to be equal. The objective is to determine if the sample means from different scenarios are statistically significantly different. The test hypotheses are:

- H_0: The means of the two samples are equal;
- H_1: The means of the two samples are not equal.

Levene's Test

Levene's test is a non-parametric test used to assess the equality of variances for a variable calculated for two or more groups. This test evaluates whether the variances of different scenarios for the same variable are equal. The test hypotheses are:

- H_0: The variances of the two samples are equal;
- H_1: The variances of the two samples are not equal.

4.8 Scenarios

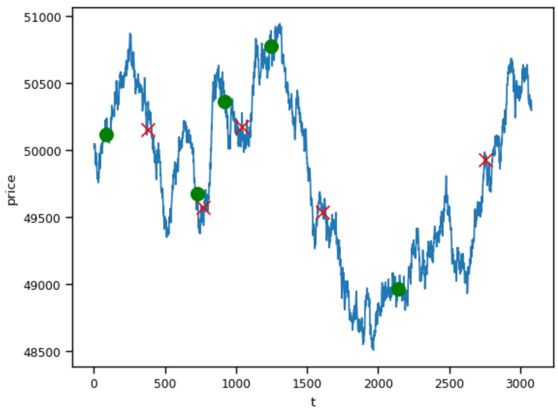
Despite the model's flexibility in charging multiple fees simultaneously, only price impact fees are evaluated herein. Analyzing the trade-offs between fixed and dynamic fees is relevant to the best fee structure study. The scenarios considered in this analysis are:

1. Fixed Fee: price impact rate is constant regardless of OI imbalance;
2. Dynamic Fee: the rate changes dynamically by decreasing if opening or closing a trade reduces the OI skew and increasing if opening/closing a trade amplifies the OI skew;
3. Dynamic Fee & Arbitrageurs: in addition to the previous scenario, this includes Arbitrageur traders who profit from opening/closing trades reducing the OI skew.

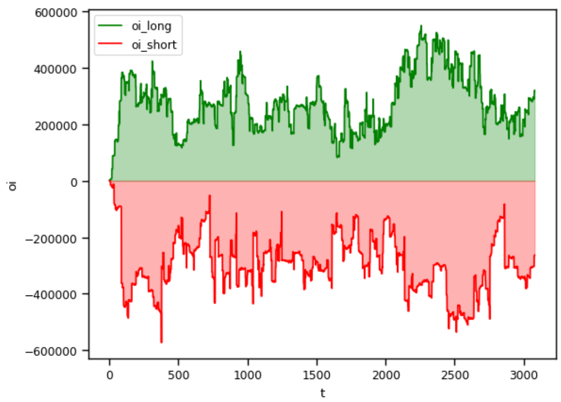
In Appendix D, all the model parameters and the scenario-specific parameters are detailed comprehensively.

4.9 Example: Single Run

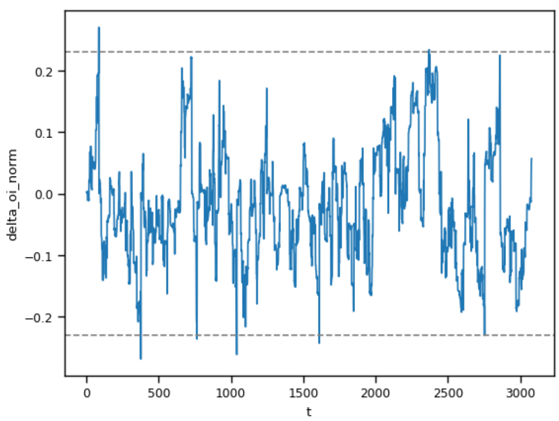
Figure 4.4 illustrates a single run in scenario 3 for a specific price path. In Figure 4.4a, arbitrageurs open and close positions at given prices, indicated by green dots and red crosses, respectively. Figure 4.4b shows the evolution of long and short open interest over time. In Figure 4.4c, the normalized open interest is depicted, with the arbitrageurs' thresholds represented by the grey line. Figure 4.4d breaks down the PnL into gross PnL and fees.



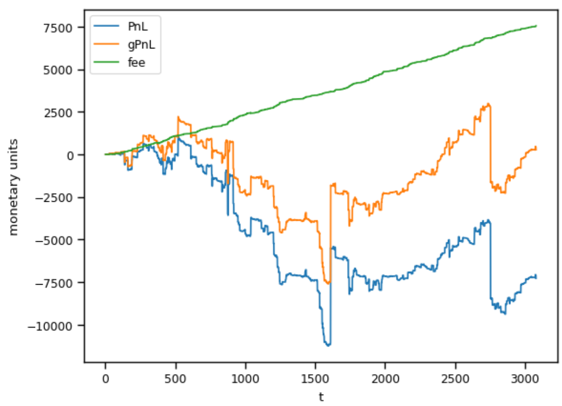
(a) Price and Arbitrageur's open (green dot) and close (red cross) orders



(b) Long and Short Open Interest



(c) Normalized OI skew (ΔOI^{norm}) and Arbitrageur Threshold levels (grey)



(d) Cumulative PnL, gPnL, and fee

Figure 4.4: Example of single run with dynamic price impact fee and arbitrageurs

Chapter 5

Results

In this chapter, we present a comprehensive analysis of the performance of different trading scenarios by evaluating key metrics, proposed in Chapter 4. Our primary focus is on the PnL figures, which are crucial for assessing the effectiveness of each scenario for liquidity providers. By examining various statistical tests and detailed breakdowns of PnL components, we aim to identify which scenario offers the most advantageous outcomes for LPs and understand the underlying factors driving these results.

5.1 Value at Risk

Regarding PnL, negative or small values are beneficial for the vault, indicating that a scenario with a lower PnL is more favorable for the LPs.

Figure 5.1 depicts the Value at Risk (VaR) across different scenarios at varying confidence intervals. Scenario 3 consistently shows the lowest VaR compared to the other scenarios. In Scenario 3, VaR increases linearly as the confidence interval rises. In contrast, both Scenario 1 and Scenario 2 exhibit higher VaR values than Scenario 3, with a non-linear increase as the confidence interval increases. This non-linear trend indicates a more significant impact of extreme outcomes in Scenarios 1 and 2. This indicates that dynamic price impact fees and arbitrageurs reduce the directional exposure.

Although this metric strongly suggests that scenario 3 is superior to the others, it is crucial to analyze the individual components of the PnL and perform statistical tests to determine the statistical significance and effectively distinguish between scenarios.

5.2 PnL Breakdown

Performing the Kolmogorov-Smirnov test on variables in different scenarios helps determine if they come from a normal distribution. Variables sum_PnL, sum_gPnL, and sum_fee for scenarios 1, 2, and 3 do not reject the null hypothesis (H_0): "the sample comes from a normal distribution." Thus, it is acceptable to use the mean and standard deviation to compare samples.

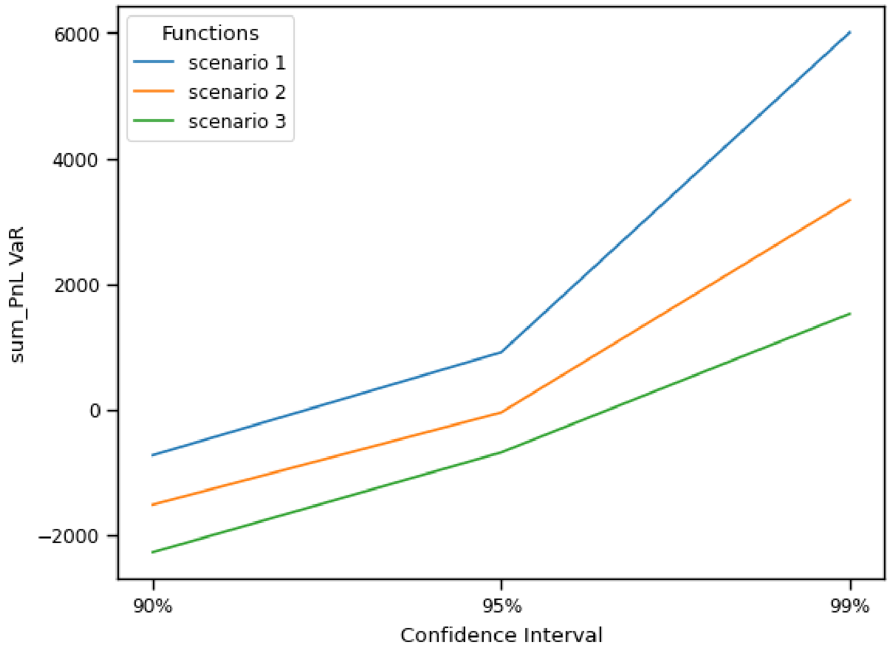


Figure 5.1: Value at Risk (VaR) with 90%, 95% and 99% Confidence Intervals for sum_PnL in Scenario 1, 2, and 3

5.2.1 PnL

Figure 5.2 graphically represents the characteristics of sum_PnL for each scenario. Scenario 1 sum_PnL has a mean of -7779 and a standard deviation of 5709. Scenario 2 sum_PnL has a mean of -8596 and a standard deviation of 5915. Scenario 3 sum_PnL has a mean of -7208 and a standard deviation of 3981.

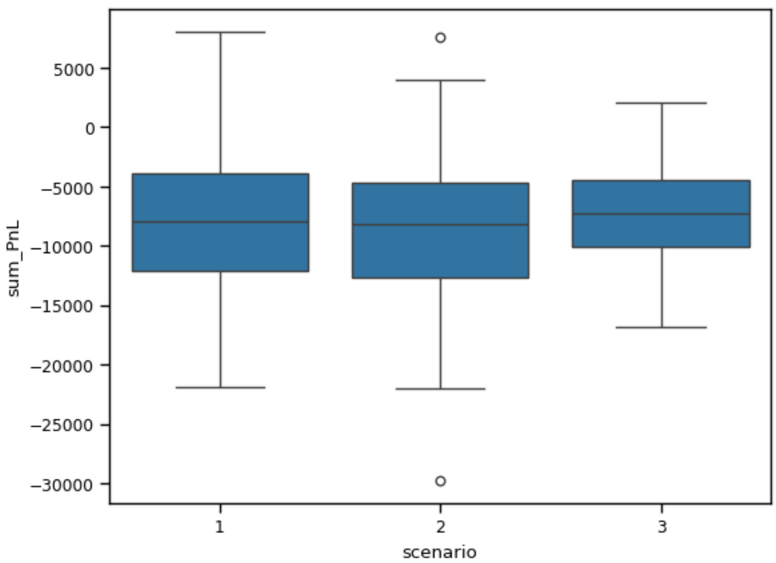


Figure 5.2: PnL distribution by sceario

In the Welch's t-test, the null hypothesis is rejected only in the mean comparison between scenario 1 and 2. The mean sum_PnL in scenario 2 is statistically different from scenario 3.

Levene's variance test indicates that scenario 3 variance is different from scenarios 1 and 2, suggesting that the dynamic price impact fee and arbitrageurs combination reduces the uncertainty in the sum_PnL, indicating a decrease in the directional exposure for PnLs.

To understand the source of these changes in means and variances from different scenarios, the PnL components (gPnL and fee) are analyzed below.

5.2.2 Gross PnL

The sum_gPnL represents the trader's PnL before fees. This variable is only influenced by traders' behavior, and the standard deviation is linked with directional exposure. Figure 5.3 represents the scenario sample distribution of sum_gPnL. Scenario 1 has a mean of 84 and a standard deviation of 5732. Scenario 2 has a mean of -801 and a standard deviation of 5925. Scenario 3 has a mean of 586 and a standard deviation of 4029.

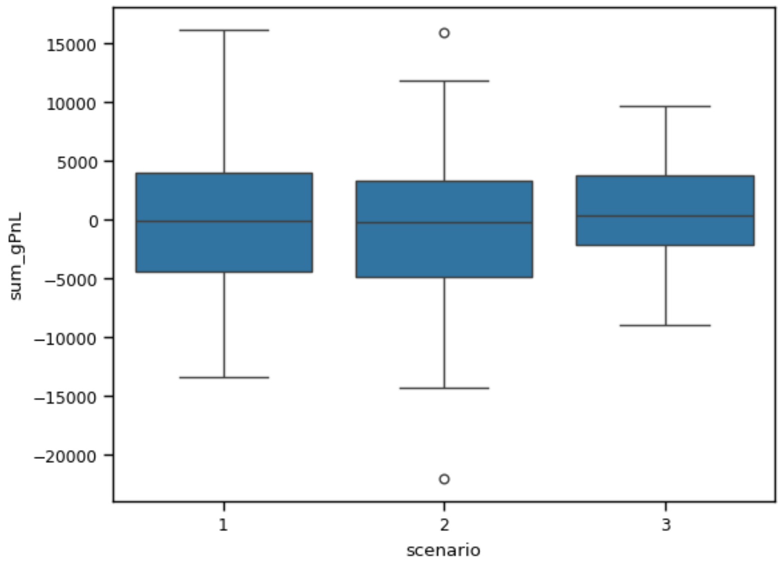


Figure 5.3: Gross PnL distribution by scenario

The conclusions on the mean and standard deviation are the same as for sum_PnL, where the mean in scenarios 2 and 3 is different and scenario 3's variance is different from scenarios 1 and 2.

Theoretically, the sample mean and standard deviation in scenarios 1 and 2 should be the same because the difference in fee structure does not affect trading behavior, suggesting that randomness in the model generated this noise in the results.

The decrease in variance for scenario 3 sum_gPnL indicates that the dynamic price impact fee and arbitrageurs combination reduces the deviation in the trader's PnL that comes only from price variation.

5.2.3 Fee

Figure 5.4 represents the sample distribution of total fees charged in different scenarios per run. Scenario 1 has a mean of 7863 and a standard deviation of 359. Scenario 2 has a mean of 7796 and a standard deviation of 382. Scenario 3 has a mean of 7885 and a standard deviation of 396.

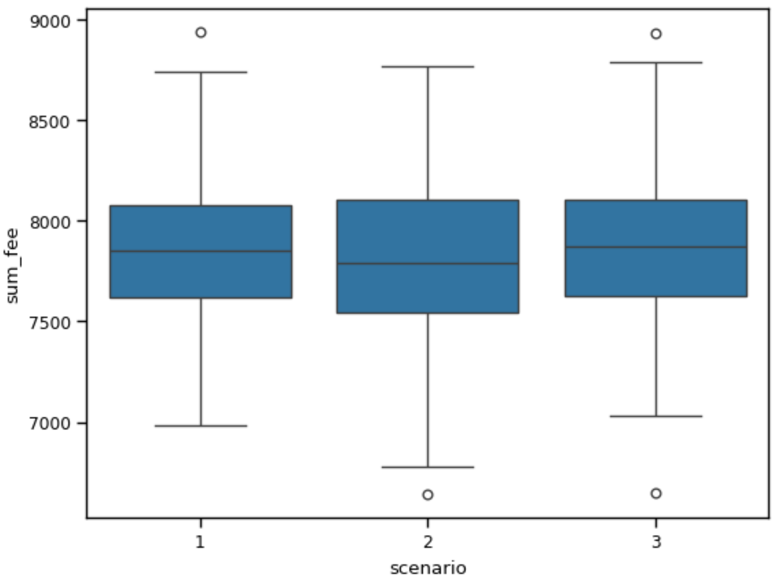


Figure 5.4: Fee distribution by scenario

In Welch's t-test, the sum_fee mean in scenarios 2 and 3 is statistically different. Scenarios 1, 2, and 3 have equal variance for variable sum_fee, as Levene's test of the null hypothesis was not rejected in any of the cases.

For fees in scenario 3, there are two distinct groups of traders: speculators and arbitrageurs. Figure 5.5 shows the speculators' sum of fees, sum_fee_sp, which does not include arbitrageurs. The mean and standard deviation in scenarios 1 and 2 do not change from all traders' fees to only speculators' fees. However, in scenario 3, arbitrageurs collect negative fees which explain the upward shift in the boxplot. For the sum_fee_sp sample, scenario 3 has a mean of 8039 and a standard deviation of 415. Regarding statistical tests, now scenario 3 does not have an equal mean to scenarios 1 and 2, whereas previously, the null hypothesis was only rejected in the t-test between scenarios 2 and 3. Scenario 3 has active arbitrageurs that profit from dynamic price impact fees. Figure 5.6 represents the distribution of fees collected by these kinds of traders. The sample mean is -155 and the standard deviation is 51.

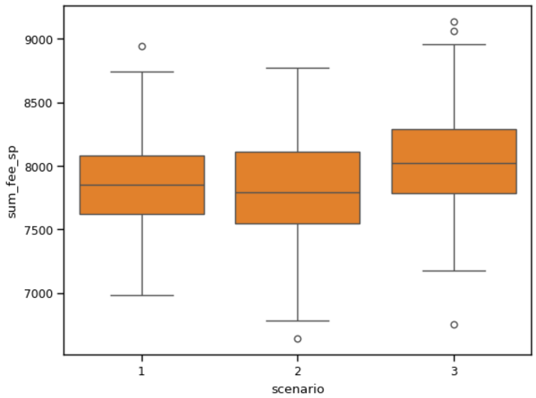


Figure 5.5: Speculator fee distribution by scenario

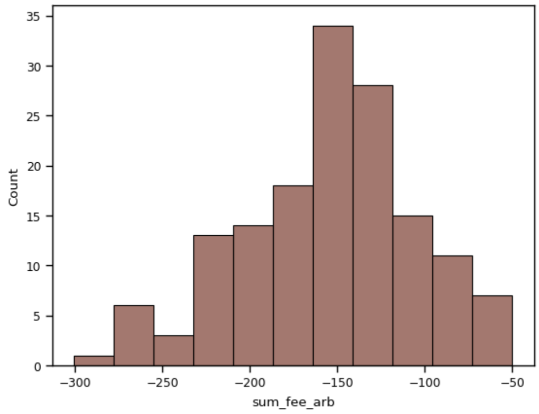


Figure 5.6: Scenario 3 arbitrageur collected fee histogram

5.2.4 OI Imbalance Score (ΔOI_{score})

The ΔOI_{score} measures the average LPs' OI imbalance per run. Figure 5.7 represents the ΔOI_{score} distribution in different scenarios. Scenario 1 has a mean of 0.03 and a standard deviation of 0.008. Scenario 2 has a mean of 0.03 and a standard deviation of 0.009. Scenario 3 has a mean of 0.01 and a standard deviation of 0.001.

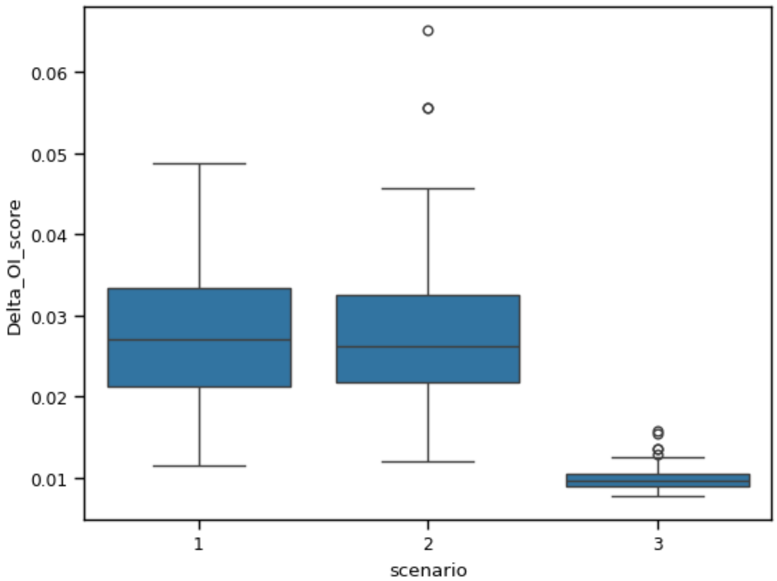


Figure 5.7: ΔOI_{score} by scenario

The variable ΔOI_{score} does not come from a normal distribution, as the null hypothesis was rejected in the K-S test. However, if we assume a normal distribution and perform the t-test, the mean ΔOI_{score} is not equal between scenario 3 and scenarios 1 and 2. Regarding variances, scenario 3 does not have equal variances when compared to scenarios 1 and 2.

This indicates that a dynamic price impact fee without arbitrageurs does not properly reduce the Open Interest imbalance.

5.3 Summary of finding

The comprehensive analysis of PnL VaR, and associated statistical tests indicates that Scenario 3 provides the most advantageous outcomes for liquidity providers. The combination of dynamic price impact fees and arbitrageurs significantly reduces risk and uncertainty, leading to more stable and predictable performance. This scenario not only mitigates directional exposure but also ensures a balanced fee structure and open interest distribution, making it the optimal choice for LPs.

Scenario 1 is distinctly different from Scenario 3, showing higher risks and less favorable PnL metrics. Although Scenario 2 is similar to Scenario 1 in many aspects, the presence of arbitrageurs in Scenario 3 makes a significant difference. The arbitrageurs contribute to reducing risk and improving PnL stability, highlighting the effectiveness of their inclusion in the trading scenario.

Chapter 6

Conclusion

The existing literature does not cover the fees associated with oracle pricing protocols in much detail, and there is no standardized nomenclature for these fees, making it difficult for new traders to understand multiple exchanges. This thesis proposed a standardized nomenclature for categorizing fee structures across different protocols, which can serve as a foundation for systematic comparisons. The detailed comparison between protocols like GMX, gTrade, and Avantis highlighted significant differences and similarities in their fee mechanisms, offering a clearer understanding of how each protocol manages risk and incentivizes liquidity. This standardized approach to nomenclature and fee comparison is crucial for developing best practices and improving overall transparency and user experience in the DeFi ecosystem.

This thesis has investigated the complexities and intricacies of fee structures in oracle pricing perpetual futures decentralized exchanges (DEXs) using an agent-based simulation approach. The analysis provided insights into how different fee structures impact trader behavior, liquidity provider (LP) profitability, and overall market stability. The findings have significant implications for the design and implementation of efficient and effective fee mechanisms in decentralized finance (DeFi) platforms.

The research demonstrated that fee structures play a crucial role in shaping market dynamics. The implementation of dynamic price impact fees, in particular, showed a marked influence on the behavior of arbitrageurs and speculators. Dynamic fees helped in balancing the open interest (OI) more effectively than static fees, as they adapt to the market conditions and reduce the chances of extreme price distortions.

The use of ABS proved to be a robust method for simulating the behavior of different market participants and the resulting market outcomes. By modeling speculators, arbitrageurs, and liquidity providers, the study captured the complex interactions and feedback loops that characterize decentralized exchanges. The ABS approach provided a detailed understanding of how fee structures affect individual trader strategies and overall market efficiency.

The evaluation metrics, including Profit and Loss (PnL), Value at Risk (VaR), and OI imbalance score, offered comprehensive measures of the impact of different fee structures. Scenario 3, which combined dynamic price impact fees with active arbitrageurs, resulted in the lowest variance in PnL and the most stable market conditions. This scenario minimized directional exposure and volatility, thereby enhancing the predictability and stability of the trading environment.

Statistical tests such as the Kolmogorov-Smirnov test, Welch's t-test, and Levene's test confirmed the significance of the findings. The results showed that dynamic fee structures significantly differ in performance compared to static structures, with dynamic fees providing better outcomes in terms of reduced risk and enhanced market stability.

The findings of this thesis have practical implications for the design of decentralized exchanges in the DeFi space. Implementing dynamic fee structures can lead to more efficient and stable markets by aligning trader incentives with market conditions. This, in turn, can attract more participants, enhance liquidity, and reduce the systemic risks associated with extreme market movements.

While this study has provided valuable insights, it also opens up several avenues for future research. Future studies could explore the impact of different types of dynamic fee structures, such as those based on volatility or liquidity metrics. Additionally, extending the ABS framework to include more diverse trader profiles and market conditions could provide a deeper understanding of the robustness of various fee mechanisms.

Future research should extend the trading model to consider a broader range of assets, improving the realism of trader behavior simulation by using advanced tools that better mimic actual trading behaviors. Additionally, experiments should be conducted with various fee structures applied simultaneously to identify the optimal configurations. It is also essential to explore different trading scenarios to determine if a particular fee structure consistently outperforms others or if its effectiveness depends on market conditions, such as low volatility versus high volatility environments. This comprehensive approach will provide deeper insights into fee optimization and enhance the robustness and efficiency of decentralized finance platforms.

In conclusion, this thesis has highlighted the critical role of fee structures in the functioning of oracle pricing perpetual futures DEXs. The use of agent-based simulation has enabled a detailed analysis of market dynamics and the effectiveness of different fee mechanisms. The findings underscore the benefits of dynamic fee structures in creating more stable and efficient markets. By aligning fees with market conditions, DeFi platforms can enhance their resilience and attractiveness to a broader range of participants. This research contributes to the ongoing efforts to improve the design and implementation of decentralized financial systems, paving the way for more robust and scalable DeFi ecosystems.

Bibliography

Sirio Aramonte, Wenqian Huang, and Andreas Schrimpf. Trading in the defi era: automated market-maker. 2021.

R. E. Bailey. A History of Money: From Ancient Times to the Present Day. *The Economic Journal*, 105(430):763–765, 05 1995. ISSN 0013-0133. doi: 10.2307/2235045. URL https://doi.org/10.2307/2235045.

Andrea Barbon and Angelo Ranaldo. On the quality of cryptocurrency markets: Centralized versus decentralized exchanges, 2023. URL https://arxiv.org/abs/2112.07386.

Binance Futures. https://www.binance.com/en/futures/home, 2024.

Chaos Labs. https://chaoslabs.xyz/, 2024.

Erdong Chen, Mengzhong Ma, and Zixin Nie. Perpetual future contracts in centralized and decentralized exchanges: Mechanism and traders' behavior. *Electronic Markets*, 34(1):35, June 2024. ISSN 1422-8890. doi: 10.1007/s12525-024-00715-1. URL https://doi.org/10.1007/s12525-024-00715-1.

DeFiLlama. Dexs derivatives volume and defi tvl. https://defillama.com/derivatives, 2024.

Omer Goldberg, Shai Kritz, and Yonatan Haimovich. Gmx v2 genesis risk framework and methodology. 2023a.

Omer Goldberg, Shai Kritz, Franziska Wehrmann, and Barry Fried. Avantis finance genesis parameter recommendations. 2023b.

R.L. Harrison. Introduction to monte carlo simulation. *AIP Conference Proceedings*, 1204:17–21, January 2010. doi: 10.1063/1.3295638.

Adam Hayes, Samantha Silberstein, and Vikki Velasquez. Perpetual futures: What they are and how they work. https://www.investopedia.com/what-are-perpetual-futures-7494870, 2023.

Siti Nur Iqmal Ibrahim, Masnita Misiran, and Mohamed Faris Laham. Geometric fractional brownian motion model for commodity market simulation. *Alexandria Engineering Journal*, 60(1): 955–962, 2021. ISSN 1110-0168. doi: https://doi.org/10.1016/j.aej.2020.10.023. URL https://www.sciencedirect.com/science/article/pii/S111001682030541X.

Andrew Leelarthaepin. Contracts for difference (cfds) - an introduction. *Australian Journal of Financial Planning*, 1(2):19–24, 2006. URL https://search.informit.org/doi/10.3316/informit.609932662779939.

Nick Lioudis. What is the gold standard? advantages, alternatives, and history. https://www.investopedia.com/ask/answers/09/gold-standard.asp, 2024.

Nick Lioudis, Gordon Scott, and Amanda Jackson. How do futures contracts roll over? https://www.investopedia.com/ask/answers/073015/how-do-futures-contracts-roll-over.asp, 2021.

Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

Nico. Intro to ethereum. https://ethereum.org/en/developers/docs/intro-to-ethereum/#what-is-a-blockchain, 2023.

Ostium Labs. The imbalance score: A novel metric for rwa-focused perpetual dexes. 2024a.

Ostium Labs. Ostium documentation. https://ostium-labs.gitbook.io/ostium-docs, 2024b.

Michael Peneder. Digitization and the evolution of money as a social technology of account. *Journal of Evolutionary Economics*, 32(1):175–203, January 2022. doi: 10.1007/s00191-021-00729-. URL https://ideas.repec.org/a/spr/joevec/v32y2022i1d10.1007_s00191-021-00729-4.html.

Kaihua Qin, Liyi Zhou, Yaroslav Afonin, Ludovico Lazzaretti, and Arthur Gervais. Cefi vs. defi – comparing centralized to decentralized finance, 2021.

Nathan Reiff, Erika Rasure, and Vikki Velasquez. The collapse of ftx: What went wrong with the crypto exchange? https://www.investopedia.com/what-went-wrong-with-ftx-6828447, 2024.

Robert J Shiller. Measuring asset values for cash settlement in derivative markets: Hedonic repeated measures indices and perpetual futures. Working Paper 131, National Bureau of Economic Research, December 1993. URL http://www.nber.org/papers/t0131.

The Block. Cexs futures volume. https://www.theblock.co/data/crypto-markets/futures, 2024.

Tongya Wang. *Behavioural Biases and Evolutionary Dynamics in an Agent-Based Financial Market*. Doctor of philosophy, University of Leeds, March 2014.

Sam M. Werner, Daniel Perez, Lewis Gudgeon, Ariah Klages-Mundt, Dominik Harz, and William J. Knottenbelt. Sok: Decentralized finance (defi), 2022. URL https://arxiv.org/abs/2101.08778.

Ce Yu, Xiang Chen, Chunyu Wang, Hutong Wu, Jizhou Sun, Yuelei Li, and Xiaotao Zhang. An improved platform for multi-agent based stock market simulation in distributed environment. *IEICE TRANSACTIONS on Information and Systems*, E98-D(10):1727–1735, October 2015. ISSN 1745-1361. doi: 10.1587/transinf.2015EDP7050.

Appendix A

Background

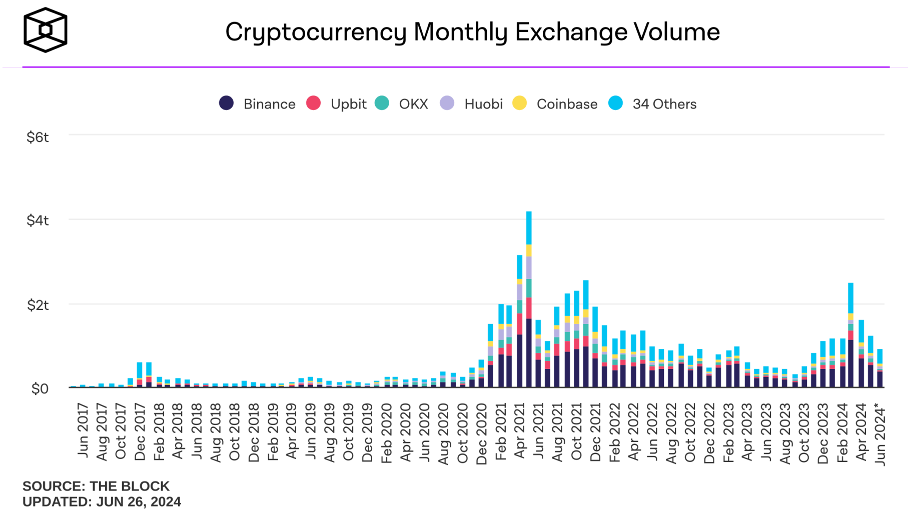


Figure A.1: Cryptocurrency Monthly Exchange Volume by The Block (2024)

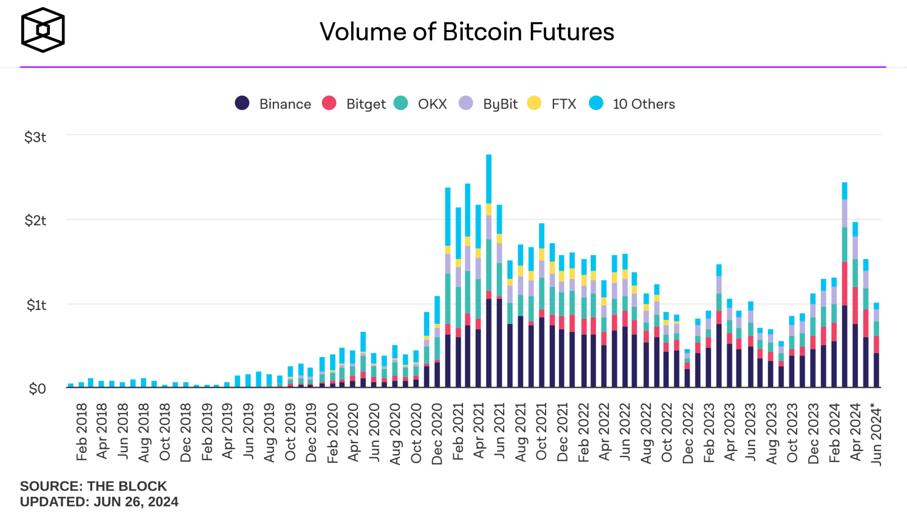


Figure A.2: Bitcoin Monthly Futures Volume by The Block (2024)

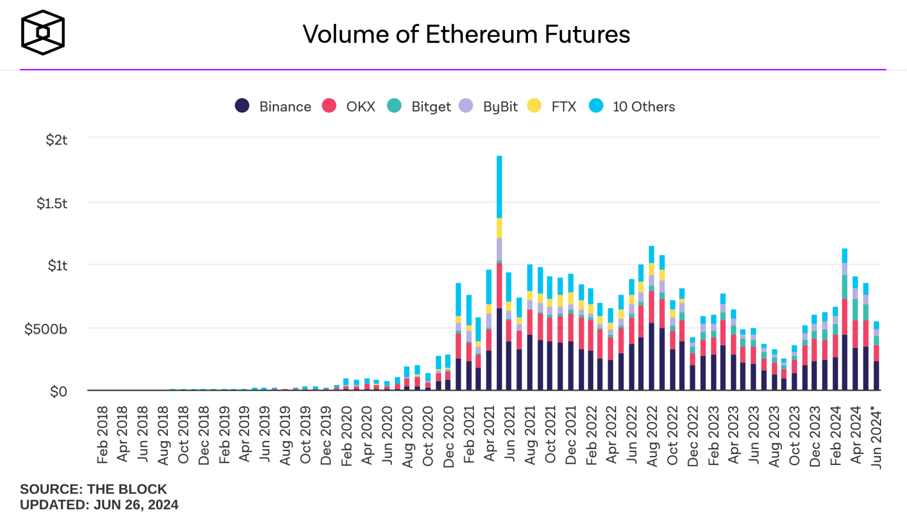


Figure A.3: Ethereum Monthly Futures Volume by The Block (2024)

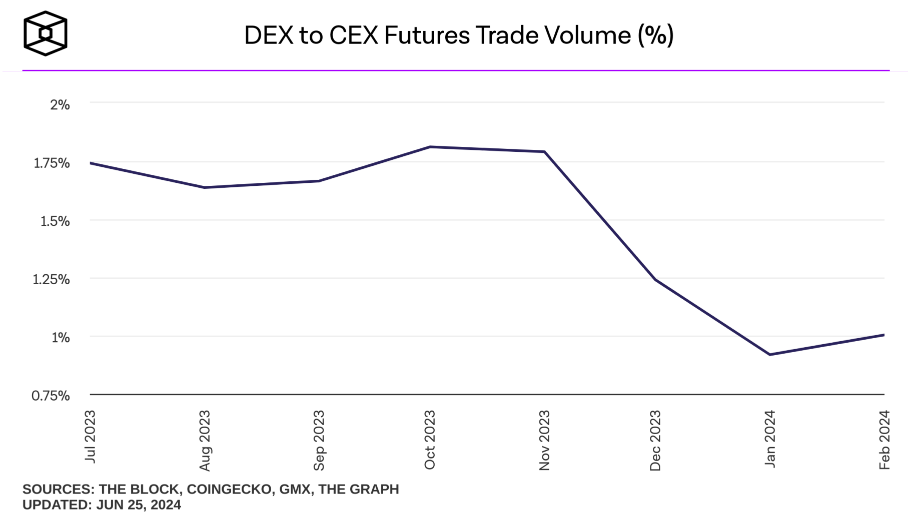


Figure A.4: DEX to CEX Futures Trade Volume by The Block (2024)

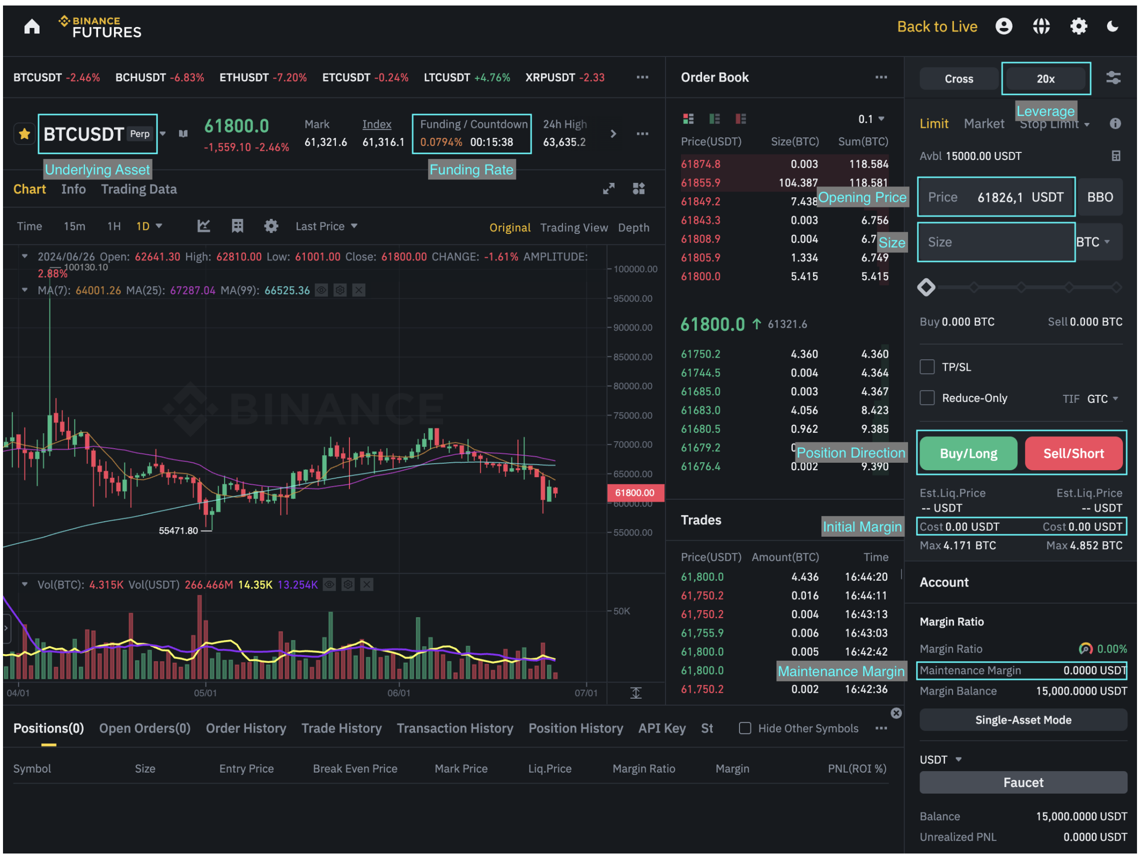


Figure A.5: Opening a Perpetual Future in Binance Futures (2024)

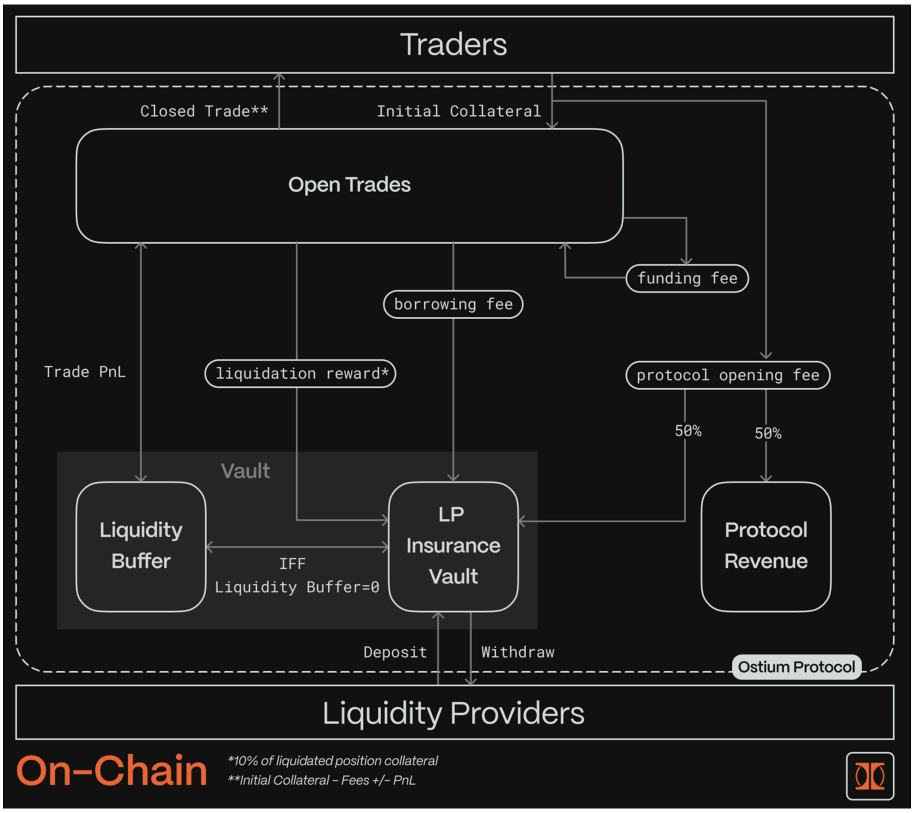


Figure A.6: Architecture diagram by Ostium Labs (2024b)

Appendix B

Traders Algorithms

Input: *price*, current period t, *short_window*, *long_window*
Output: Trading signal
if $length(price[:t]) < long\_window$ **then**
  return 0 `// no trading signal`
end
$short\_ma \leftarrow \frac{1}{short\_window} \sum_{i=t-short\_window}^{t} price[i]$;
$long\_ma \leftarrow \frac{1}{long\_window} \sum_{i=t-long\_window}^{t} price[i]$;
if $short\_ma > long\_ma$ **then**
  return 1 `// long trading signal`
else
  $short\_ma < long\_ma$
end
return-1 `// short trading signal`

Algorithm 2: Moving Average Crossover Strategy

Input: *price*, current period t, *window*, *threshold*
Output: Trading signal
if $length(price[:t]) < window$ **then**
  return 0 `// no trading signal`
end
$mean\_price \leftarrow \frac{1}{window} \sum_{i=t-window}^{t} price[i]$;
$std\_price \leftarrow \sqrt{\frac{1}{window} \sum_{i=t-window}^{t} (price[i] - mean\_price)^2}$;
if $price[t] < mean\_price - threshold \cdot std\_price$ **then**
  return 1 `// long trading signal`
else if $price[t] > mean\_price + threshold \cdot std\_price$ **then**
  return-1 `// short trading signal`
else
  return 0 `// no trading signal`
end

Algorithm 3: Mean Reversion Strategy

Input: $price$, current period t, $window$, $buy\_threshold$, $sell\_threshold$
Output: Trading signal
if $length(price[:t]) \leq window$ **then**
  return 0 `// no trading signal`
end
$roc \leftarrow \frac{price[t]-price[t-window]}{price[t-window]}$;
if $roc > buy\_threshold$ **then**
  return 1 `// long trading signal`
else if $roc < sell\_threshold$ **then**
  return-1 `// short trading signal`
else
  return 0 `// no trading signal`
end

Algorithm 4: Momentum Strategy

Input: Probability of going long p_long
Output: Trading signal
$rand \leftarrow random\_value \in [0,1]$ `// Generate a random value between 0 and 1`
if $rand < p\_long$ **then**
  return 1 `// long trading signal`
else
  return-1 `// short trading signal`
end

Algorithm 5: Random Position Strategy

Input: $threshold$, $\Delta OI(t)$
Output: Boolean indicating if a position should be opened
if $threshold \leq \Delta OI(t)$ **then**
  return True `// Open position`
else
  return False `// Do not open position`
end

Algorithm 6: Step to open arbitrageur position

Input: $\Delta OI(t)$, OI_{cap}
Output: Set position details
if $\Delta OI(t) > 0$ **then**
  $long \leftarrow False$;
else
  $long \leftarrow True$;
end
$leverage \leftarrow 1$;
$collateral \leftarrow |\Delta OI(t)| \cdot OI_{cap}$;

Algorithm 7: Get arbitrageur position details

```
Input: ΔOI(t), D_i, threshold
Output: Boolean indicating if a position should be closed
if D_i = 1 then
    if ΔOI(t) > threshold then
        return True // Close position
    end
if D_i = −1 then
    if ΔOI(t) < −threshold then
        return True // Close position
    end
else
    return False // Do not close position
end
```

Algorithm 8: Step to close arbitrageur position

Appendix C

Fees Examples

C.1 Order Execution Rates

$$EF_i = gPV_i(t_i^{open}) \cdot \tau_i^{open} + gPV_i(t_i^{close})$$

In $\tau_{open}\tau_{close}$ are the opening and closing rates for order execution fee.

Fixed fee

$$\tau_i^{open} = \tau_i^{close} = k, \forall i \in I \tag{C.1}$$

$$\tag{C.2}$$

Dynamic

Measuring the impact on the Delta norm impact when the position is opened:

$$\delta_i^{open} = \frac{|\Delta OI(t_i^{open}) + L_i C_i D_i| - |\Delta OI(t_i^{open})|}{OI_{cap}}, \delta_i^{open} \in [-1, 1] \tag{C.3}$$

$$\delta_i^{close} = \frac{|\Delta OI(t_i^{close}) - L_i C_i D_i| - |\Delta OI(t_i^{close})|}{OI_{cap}}, \delta_i^{close} \in [-1, 1] \tag{C.4}$$

$$\tau_i^{open} = k(1 + \alpha \cdot \delta_i^{open}), \forall i \in I \tag{C.5}$$

$$\tau_i^{close} = k(1 + \alpha \cdot \delta_i^{close}), \forall i \in I \tag{C.6}$$

C.2 Rolling Rate

$$RF_i = \sum_{\substack{t \in T_i \\ t \neq t_i^{open}}} gPV_i(t) \cdot \phi_i(t)$$

Velocity-Based Funding rate

$$fr(t) = fr(t-1) + k \cdot \Delta OI^{norm}(t), \forall t \in T \tag{C.7}$$

Charged on the dominant side.

$$fr_{long}OI_{long} = -fr_{short}OI_{short} \tag{C.8}$$

$$fr_{long}(t) = \begin{cases} |fr(t)| & \text{if } fr(t) > 0 \\ -|fr(t)|\dfrac{OI_{short}}{OI_{long}} & \text{if } fr(t) < 0 \end{cases} \tag{C.9}$$

$$fr_{short}(t) = \begin{cases} -|fr(t)|\dfrac{OI_{long}}{OI_{short}} & \text{if } fr(t) > 0 \\ |fr(t)| & \text{if } fr(t) < 0 \end{cases} \tag{C.10}$$

$$\phi_i(t) = \begin{cases} fr_{long}(t) & \text{if } D_i = 1 \\ fr_{short}(t) & \text{if } D_i = -1 \end{cases} \tag{C.11}$$

Borrowing Rate

$$\phi_i(t) = \begin{cases} k\Delta OI^{norm}(t) & \text{if } D_i \cdot \Delta OI(t) > 0 \\ 0 & \text{if } D_i \cdot \Delta OI(t) <= 0 \end{cases} \tag{C.12}$$

C.3 Price Impact Rate

$$PF_i = L_iC_i\frac{P(t_i^{close})}{P(t_i^{open})}\frac{\zeta^{open} + \zeta^{close}}{1 + D_i \cdot \zeta^{open}} \tag{C.13}$$

Fixed

$$\zeta_i^{open} = \zeta_i^{close} = k, \forall i \in I \tag{C.14}$$

Dynamic

$$\zeta_i^{open} = k(1 + \alpha \cdot \delta_i^{open}) \tag{C.15}$$

$$\zeta_i^{close} = k(1 + \alpha \cdot \delta_i^{close}) \tag{C.16}$$

Appendix D

Simulation Model Parameters

Price parameters are the inputs for Geometric Brownian Motion with constant drift and volatility.

Table D.1: Price Parameters

Name	**Value**	**Units**	**Range**	**Description**
P_0	50000	\$	$\mathbb{R}^+$	Starting price at time $t = 0$
μ_{price}	0	minute	$\mathbb{R}$	Mean rate of return (drift coefficient)
σ_{price}	0.0007	minute	$\mathbb{R}^+$	Standard deviation of returns
T	2880	minute	$\mathbb{I}$	The total time period over which the simulation is run

The initial price is set to match the current market price of Bitcoin, providing a realistic starting point for the simulation. The mean rate of return is set to zero because the simulation considers a two-day period, during which no significant directional movement is assumed. This neutral stance reflects an expectation of balanced market forces over the short term. The standard deviation is derived from analyzing Bitcoin's one-minute price data over two days, with the median value selected to represent typical market volatility. The total time period, T, is measured in minutes, a common unit for financial simulations, encompassing two full days plus an additional 200 periods to capture extended market dynamics. This choice ensures a detailed and granular representation of price movements, essential for robust financial modeling.

Table D.2: Speculator Parameters

Name	Value	Units	Range	Description
λ_{open}	0.625	1/minute	$\mathbb{R}^+$	Opening trades average rate
$1/\lambda_{hold}$	90	minutes	$\mathbb{R}^+$	Holding periods average rate
μ_{lev}	10	units	$\mathbb{R}^+$	Mean in the leverage normal distribution
σ_{lev}	3	units	$\mathbb{R}^+$	Standard deviation in the leverage normal distribution
min_lev	2	units	$\mathbb{R}^+$	Minimum leverage
max_lev	20	units	$\mathbb{R}^+$	Maximum leverage
shape_col	6.233	monetary unit	$\mathbb{R}^+$	Shape in the collateral lognormal distribution
location_col	200	monetary unit	$\mathbb{R}^+$	Location in the collateral lognormal distribution
scale_col	1.55	monetary unit	$\mathbb{R}^+$	Scale in the collateral lognormal distribution
min_col	100	monetary unit	$\mathbb{R}^+$	Minimum collateral
max_col	5000	monetary unit	$\mathbb{R}^+$	Maximum collateral
MAC_short_window	50	time	$\mathbb{N}$	Short window in Moving Average Crossover strategy
MAC_long_window	200	time	$\mathbb{N}$	Long window in Moving Average Crossover strategy
MR_window	20	time	$\mathbb{N}$	Window in mean reversion strategy
MR_threshold	2	-	$\mathbb{R}^+$	Threshold in mean reversion strategy
M_window	50	time	$\mathbb{N}$	Window in momentum strategy
M_buy_threshold	0.01	-	[0,1]	Buy threshold in momentum strategy
M_sell_threshold	-0.01	-	[0,1]	Sell threshold in momentum strategy
R_p_long	0.5	-	[0,1]	Long probability in random strategy

Table D.3: Arbitrageur Parameters

Name	Value	Units	Range	Description
arb_active	0	boolean	{TRUE, FALSE}	0: not active, 1: active
arb_threshold	0.23	adimensional	[0,1]	Arbitrageur threshold to open/-close position

Table D.4: Fee structure and Parameters

Category	**Timing**	**Structure**
Order Execution	open close	$\tau_i^{open} = 0$ $\tau_i^{close} = 0$
Rolling	holding	$\phi_i(t) = 0$
Price Impact	open close	$\zeta_i^{open} = k_{PF}(1 + \alpha_{PF} \cdot \delta_i^{open})$ $\zeta_i^{close} = k_{PF}(1 + \alpha_{PF} \cdot \delta_i^{close})$

Table D.5: Specific-scenario parameters

	Scenario		
Parameter	**1**	**2**	**3**
k_{PF}	0.025%	0.025%	0.025%
α_{PF}	0	5	5
arb_active	FALSE	FALSE	TRUE

Appendix E

Results Summary Metrics and Statistical Tests

Table E.1: Result sample mean and standard deviation

	sum_PnL		sum_gPnL		sum_fee		sum_fee_sp		Delta_OI_score	
Scenario	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
1	-7779	5709	84	5732	7863	359	7863	359	0.03	0.008
2	-8596	5915	-801	5925	7796	382	7796	382	0.03	0.009
3	-7298	3981	586	4029	7885	396	8039	415	0.01	0.001

Table E.2: Statistical tests p-values

Test	**Scenario**	**sum_PnL**	**sum_gPnL**	**sum_fee**	**sum_fee_sp**	**Delta_OI_score**
Kolmogorov Smirnov (normality)	1	0.96	0.90	0.29	0.28	0.00
	2	0.16	0.20	0.94	0.94	0.00
	3	0.53	0.29	0.84	0.83	0.00
Welch's t-test (mean)	1-2	0.22	0.19	0.12	0.12	0.80
	1-3	0.40	0.38	0.62	0.00	0.00
	2-3	0.03	0.02	0.04	0.00	0.00
Levene's test (variance)	1-2	0.83	0.84	0.34	0.34	0.91
	1-3	0.00	0.00	0.29	0.09	0.00
	2-3	0.00	0.00	0.87	0.45	0.00