GAUNTLET

Gauntlet Research Report

Mechanism Design Assessment

An analysis of the THORChain Continuous Liquidity Pool and its slip-based fee

THORCHAIN

November 2020

Contents

Disclaimer

This report is for informational purposes only and does not constitute an offer to sell, a solicitation to buy, or a recommendation for any security, nor does it constitute an offer to provide investment advisory or other services by Gauntlet Networks Inc. No reference to any specific security constitutes a recommendation to buy, sell or hold that security or any other security. Nothing in this report shall be considered a solicitation or offer to buy or sell any security, future, option or other financial instrument or to offer or provide any investment advice or service to any person in any jurisdiction. Nothing contained in this report constitutes investment advice or offers any opinion with respect to the suitability of any security, and the views expressed in this report should not be taken as advice to buy, sell or hold any security. The information in this report should not be relied upon for the purpose of investing. In preparing the information contained in this report, we have not taken into account the investment needs, objectives and financial circumstances of any particular investor. This information has no regard to the specific investment objectives, financial situation and particular needs of any specific recipient of this information and investments discussed may not be suitable for all investors. Any views expressed in this report by us were prepared based upon the information available to us at the time such views were written. Changed or additional information could cause such views to change. All information is subject to possible correction. Information may quickly become unreliable for various reasons, including changes in market conditions or economic circumstances.

Part I
Background

1 Overview

The dramatic user growth of decentralized finance (DeFi) applications has been driven by the rise in usage of Automated Market Makers. DeFi protocols aim to replace custodial, centralized entities such as exchanges, brokers, and lenders with smart contract based mechanisms that economically incentivize fair and efficient trade. Automated Market Makers (AMMs) have captured significant cryptocurrency exchange market share, amassing daily trading volumes larger than those of their centralized counterparts. In late 2020, decentralized AMM venues became the dominant source of liquidity for a variety of Ethereum-based digital assets, such as Wrapped Bitcoin and the Dai Stablecoin [1]. One of the main drivers of this growth has been the search for sustainable cryptocurrency yields that exceed those of traditional financial products. Providing liquidity in decentralized exchanges has been a popular way to generate yield for digital asset investors, where liquidity providers earn trading fees for providing capital to AMMs. As of October 2020, there is over $5 billion US dollars equivalent of cryptocurrency supplied by liquidity providers of Ethereum-based decentralized exchanges. The most popular type of AMM is known as the constant function market maker (CFMM), which was first popularized by Uniswap [2, 3, 4].

In AMMs, users provide liquidity for a pair of crypto-assets to a smart contract by depositing their assets in a liquidity pool that is controlled by a smart contract. When a user deposits their assets into a liquidity pool, a synthetic asset known as a liquidity provider (LP) share is minted that acts as a voucher for the user to redeem their pro-rata share of assets and fees. The quantity of each asset deposited is determined by the price that the AMM is currently quoting to prospective traders. When trader places a market order on an AMM, the protocol takes one asset from the trader and returns a quantity of the trader's desired asset from the liquidity pool. This quantity is determined by the preservation of an invariant dependent on the quantity of assets deposited into each pool and a fee. The collected fee is distributed pro-rata to LPs — e.g. if a user has 50% of LP shares before a trade takes place, they receive 50% of the fees collected for that trade. For a formal mathematical analysis of the mechanism used in Uniswap and other CFMMs, please refer to [2, 5].

THORChain is a Uniswap-like CFMM that uses its own chain to facilitate transactions. Unlike Uniswap, THORChain can facilitate cross-chain transactions, such as those between Bitcoin and Ethereum rather than providing synthetic asset swaps (e.g. trading one Ethereum asset for another). By facilitating cross-chain swaps, THORChain is able to expand the set of assets that can be traded via a CFMM-like mechanism. In particular, assets on other chains such as Bitcoin, Cosmos, and Terra can be swapped for Ethereum ERC-20 assets. You can find more info on how THORChain works in their documentation [6] and our previous analysis of THORChain system incentives [7].

While many CFMM-based exchanges use a fixed-fee, THORChain uses a fee structure that charges more for larger trades (slip-based fee). Such a fee model has the potential to increase the fees earned by liquidity providers on exchanges that use this fee structure, while also increasing the cost to execute a trade. Ideally, the increase in fees should draw more liquidity to an exchange using a slip-based fee, and this stronger liquidity would subsequently lower the cost to trade on those exchanges. On the other hand, increasing trading fees could cause a decrease in trading volume that leads to LPs removing their assets due to low yield.

However, an analytic analysis of the particular impact of a fee model is fraught with difficulties. These difficulties occur as the financial performance of the system is dependent on the behaviors, strategies, and interactions between LPs and traders. For instance, if most traders want to execute large trades but find the slip-based fee to be larger than alternatives (e.g. centralized exchanges), then the trade volume in such a CFMM will be low. If there are not many trades executed, then LPs receive smaller fees and are liable to remove their liquidity from the CFMM pool. In order to stress test how much liquidity is present, one needs to model the risk and profit seeking behaviors of the agents in the system — the LPs and traders. Using an agent-based simulation model, we investigate how profit-maximizing LPs and traders would participate in these exchanges and find conditions under which each fee structure would be more successful.

2 Simulation Analysis

Gauntlet utilizes agent-based simulation to model the THORChain protocol and associated ecosystem of LPs and traders. Agent-based simulation has been used to study the interactions between borrowers and lenders in decentralized lending protocols [8] and Uniswap [2]. We adapt these methodologies to stress test THORChain's fee model. More information about how the Gauntlet platform works can be found in the Appendix A.

2.1 Scope of the Analysis

Three key components in THORChain are the incentive pendulum, the native currency RUNE and the continuous liquidity pool (CLP) slip-based fee. The focus of this analysis is to analyze the slip-based fee and present a comparison to the fixed fee model in isolation of the other features of THORChain. This was decided in part from having already released a prior report on the incentive pendulum [7] but moreso since the numerous cross-interactions the incentive pendulum and RUNE have on the overall trading activity present challenges in making clear attributions of agent behavior towards the novel fee structure and not the other pieces of the protocol.

In particular, the incentive pendulum creates both a RUNE denominated constraint on the total value of staked assets the system can secure along with indirect exposure to inflation rewards for liquidity providers. RUNE in turn acts as the settlement currency for all liquidity pools on THORChain along with ensuring network security through incentives and serving as the governance and inflation reward token.

The multifaceted and highly active nature of RUNE within the system pose questions as to how to accurately model the liquidity and external market dynamics of the asset. As an example, a bonder's disincentivization in RUNE to act maliciously against the network is not necessarily well parameterized by the classical assumption of a geometric Brownian motion price trajectory due to the value differential of RUNE inside and outside THORChain, the subsequent predicted illiquidity of external markets along with a general sense of altruism or belief in the network given the 1MM RUNE requirement just to run the node.

Both the valuation and associated volatility of RUNE are also not so straightforward to pinpoint. A traditional financial approach might be to take the discounted cash flows of trading fees from the pools RUNE secures but a significant portion of these fees are themselves denominated in RUNE, leading to a recursive estimate. There is also likely a nonlinearity of value with respect to quantity due to the disproportionate gains from acquiring enough RUNE to obtain early bonding rewards which also comes with the confounding issues of the market impact, slippage and potential signaling of a loss of faith in the protocol a large sale may incur.

Because of the aforementioned complications that arise in decoupling the effects of RUNE and the incentive pendulum from the slip-based fee, we ultimately removed them from our model. As both the block reward and the hub currency in a hub and spoke model, RUNE draws parallels to ETH which served a similar function in Uniswap v1 so in our simplication of THORChain which we refer to as the "CLP Exchange" there consists of a single ETH-DAI liquidity pool. In order to highlight **the effects of the slip-based fee**, we create a competitive ecosystem where this fee model is pitted head-to-head against a fixed-percentage fee model. Thus we also include a XYK exchange with an ETH-DAI liquidity pool, charging a fixed-percentage fee. [1] Keeping our observations focused on a single trading pair eliminates the added contrast of the hub and spoke and point to point network models. In doing so we better incorporate the market dynamics that drive individual agent behavior and influence overall metrics of protocol success such as market share or fee revenue.

2.2 Goals of the Analysis

While we have observed fixed-fee AMM protocols like Uniswap and Curve gain success in terms of dollar notional trading volume and liquidity, one can also observe users participating by providing liquidity deal with unpredictable returns. By charging a variable fee, THORChain hopes to allow liquidity providers to maintain profitability across a wider range of scenarios. In our analysis we try to assess this by answering the following questions:

1. Does the slip-based fee improve the expected returns for liquidity providers? Under what market conditions and user behaviors is THORChain's model successful?
    (a) How does price movement in the pooled assets affect key protocol metrics?
    (b) The slip-based fee changes with trade size and pool depth. How does variation in these factors affect system livelihood?
2. The slip-based fee is often more expensive than a fixed-fee in high value, time-sensitive circumstances. To what extent does this disincentivize arbitrageurs that keep prices on THORChain in-line with other exchanges?
3. Under what scenarios does organic demand volume favor the slip-based versus the fixed fee?

[1] In much of the documentation and literature on THORChain, the constant product exchange is referred to as XYK. This name is chosen as the convex, polynomial invariant that Uniswap uses is the trading function $\phi(x, y) = xy$ [2, 5]

Part II

Model Setup and Details

3 Model Setup

The model is comprised of a few main components:

1. A Python model of the THORChain system, focused on the the CLP fee model
2. Stochastic price trajectories for asset prices
3. Models of two external exchanges — a centralized exchange and a Uniswap-style XYK Exchange
4. A set of agents that, responding to changes in price and other agents, interact with the THORChain model and the external components

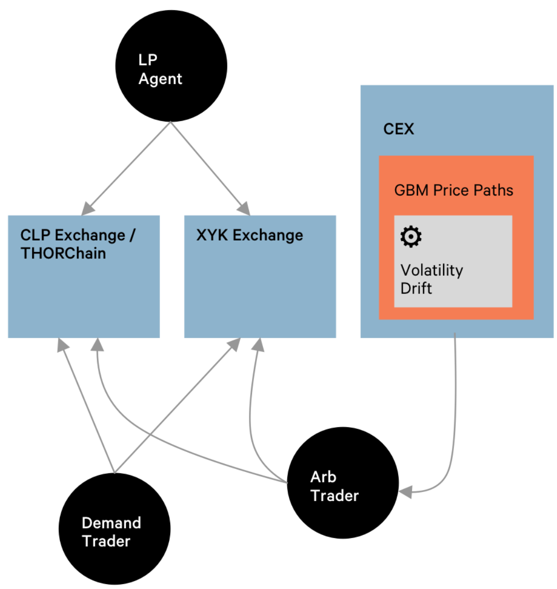

Figure 1: The Simulation Model (high-level diagram)

3.1 CLP Model

THORChain uses continuous liquidity pools (CLP) to facilitate asset swaps. Participants of the network can interact with pools in the following ways.

3.1.1 Providing Liquidity

To provide liquidity, participants provide a pair of assets to a liquidity pool and in return receive LP shares determined by

$$s = \frac{(X+Y)(xY+yX)}{4XY} \tag{1}$$

where X and Y are the pool asset balances including the newly provided asset amounts x and y [2].

3.1.2 Removing liquidity

To withdraw their liquidity, participants redeem their shares for pool assets prorata relative to the total number of LP shares outstanding.

3.1.3 Swapping

To swap, participants pay a slip-based liquidity fee on the output token to the pool given by

$$\Delta y = \frac{x^2 Y}{(x+X)^2} \tag{2}$$

and receive tokens given by

$$y = \frac{xXY}{(x+X)^2} \tag{3}$$

where X and Y are the pool input and output token balances prior to the swap and x is the amount of input tokens provided. Note that

$$y + \Delta y = \frac{xXY}{(x+X)^2} + \frac{x^2 Y}{(x+X)^2} = \frac{xY(x+X)}{(x+X)^2} = \frac{xY}{x+X} \tag{4}$$

which has the same structure as the XYK formula prior to taking the fee.

3.2 Price Feeds

One of the main driving forces in the simulation model are the synthetic price feeds which agents interact with and drive trading volumes and LP returns. We use a standard Geometric Brownian motion (GBM) to simulate price trajectories. This stochastic process obeys the Îto stochastic differential equation, $dX_t = \mu S_t dt + \sigma S_t dW_t$, where dW_t is the standard Wiener measure on $\mathbb{R}$. GBM is also equivalent to the exponential of a randomly varying quantity following a Brownian motion, e.g. $X_t = X_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right)$.

3.3 Exchanges

To test the viability of THORChain, we allow agents to provide liquidity as well as trade on two other exchanges. Agents choosing to participate in the other exchanges allows us to model the competitive landscape that THORChain will operate in. We model agents trading and providing liquidity for two assets — one volatile and one not. We've chosen ETH and DAI, as those are a popular trading pair on many decentralized exchanges. The exchanges are initialized to support these two assets.

[2] This formula has since been modified in the protocol due to issues related to asymmetric liquidity provision. As a simplification in the model that also avoids these problems all LPing is required to be symmetric.

3.3.1 Fixed-Fee XYK Exchange

The fixed-fee is set to thirty basis points (0.30%), which is a common value used on decentralized exchanges, including Uniswap [9] and Synthetix [10]. Agents are able to execute trades and provide liquidity to this exchange. We initialize the liquidity pool to a size that we vary across simulation runs, which is specified in the outputs of Part III and in section 7

3.3.2 Centralized Exchange

The simulation emulates a centralized exchange that allows agents to buy and sell ETH and DAI in exchange for USD. It uses slippage models to represent the price impact of agent trades in the simulation environment, where larger trade sizes will incur more slippage. More specifically, the slippage s is defined as

$$s = c\sigma x^{\alpha} \tag{5}$$

where c is a scalar for intensity, σ is the annualized volatility, α controls the concavity of the curve, and x is defined as

$$x = \min\left(\frac{z}{d}, 1\right) \tag{6}$$

where z is the order size and d is the average daily traded volume.

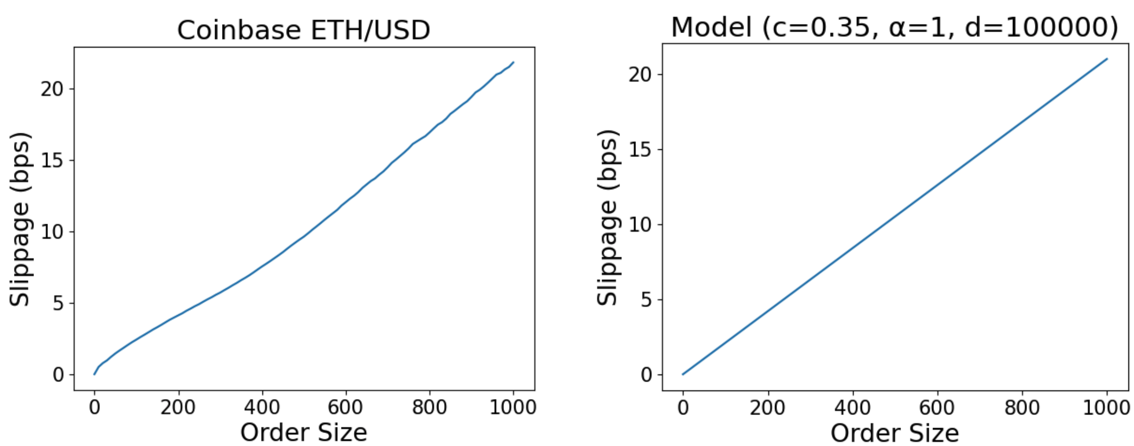

Figure 2: Coinbase ETH/USD Slippage

Using order book data from Coinbase Pro for ETH/USD and DAI/USD in Oct 2020 to fit these models we choose $c = 0.35$, $\alpha = 1$ and $d = 100000$ for ETH and $c = 0.1$, $\alpha = 0.5$ and $d = 1500000$ for DAI.

4 Agent Types

4.1 Liquidity Provider

The main decision that is modeled involves LP agents allocating funds across the XYK exchange and THORChain. Agents select whether or not to provide liquidity based on the expected returns of doing so. The agents are initialized with different "lookback windows". Lookback windows determine the timescale that agents take

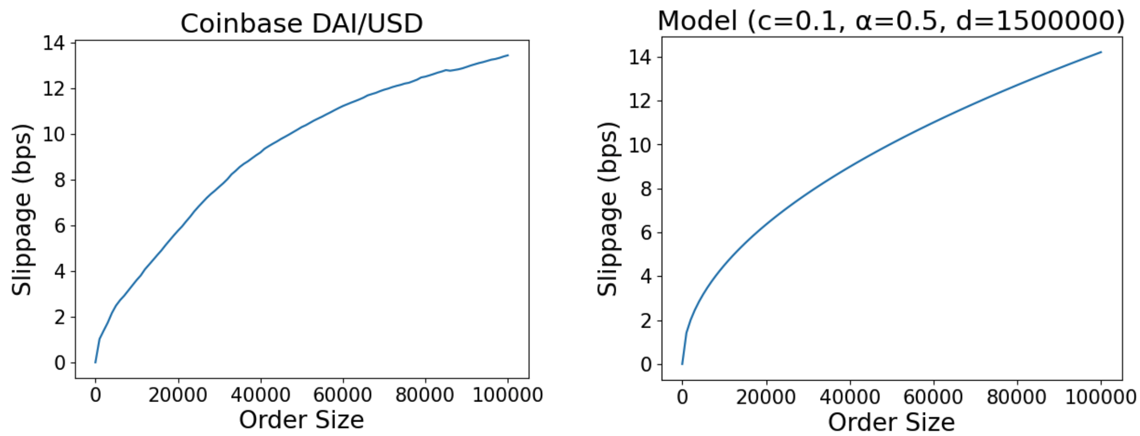

Figure 3: Coinbase DAI/USD Slippage

into consideration when making the decision to provide liquidity. Agents with a shorter lookback window calculate the expected ROI based off of more recent data, and will make decisions more frequently. When fees (and consequently, expected ROI) on one exchange start to diminish, agents with a short lookback window will stop immediately, whereas agents will a longer time-preference will wait to see if the lowered yields persist before removing their capital. As mentioned in 2.1, modeling LP decision making is fraught with hard-to-verify assumptions, so we keep our LP agent logic relatively simple. However, the ROI metrics we expose should provide a window into the decision LPs face. These statistics demonstrate under different conditions how LPs on each exchange will be incentivized.

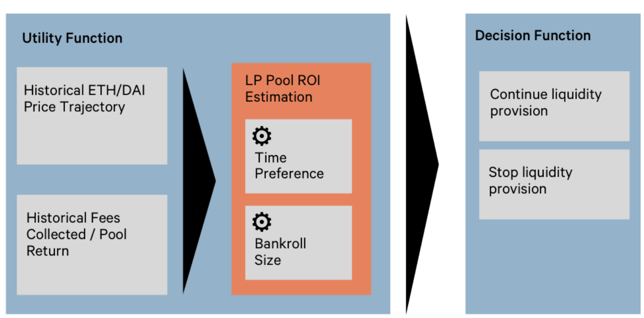

Figure 4: LP Agent Logic

4.2 Arbitrageur

Arbitrageur agents execute trades between the centralized and decentralized exchange when price differs by an amount large enough that they can realize an immediate profit.

1. This keeps the price implied by the DEX exchange pools (CLP and XYK) in line with the centralized exchange price

2. These agents are initialized with a fixed, pre-determined bankroll, which allows for realistic limits on how large of an arbitrage opportunity can be taken at one time. We note that in cryptocurrency trading, unlike traditional finance, arbitrage is often constrained by a dearth of liquidity.

One thing to note is that with a slip-based fee, arbitrageurs can try to maximize their revenue by splitting up their trades into smaller chunks to minimize slippage. This is distinct from the path deficiency property that is endemic to constant function market makers like Uniswap [2, 5]. THORChain has a mechanism called a ”swap queue” which orders transactions by slip, so that the most aggressive arbitrageur (who attempts to close the largest portion of the arbitrage opportunity) has their trade selected. We approximate a competitive market in the swap queue with our arbitrageur logic, which directs the arb agents to try to swap as much as they can.

Since arbitrageurs generally prefer to be market delta neutral, the precise sequence of trades is

1. Buy asset A on centralized exchange
2. Swap asset A for asset B on cheapest to execute DEX
3. Sell asset B on centralized exchange

As such their accounts only consist of USD at the start and end of every time step. At each decision point the arbitrageur enumerates through all sequences of possible trades taking slippage, fees and market prices from the various exchanges into consideration and executes the one with highest expected profit.

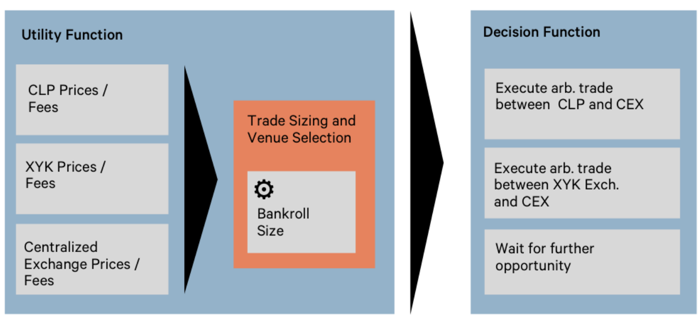

Figure 5: Arbitrage Agent Logic

4.3 Demand trader

The demand trader agent represents the demand generation process for cryptoassets simulated within the system. Agents always choose the best price for executing available trades between the exchanges in the simulation. Each agent trades a size that is fixed during a given simulation run; however, this trade size is varied across simulation scenarios. The choice of trade size parameter is critical as it can have a large affect on the simulation results. In traditional finance, market microstructure models are often specified completely by the trade size distribution and we mimic such models in our parametrization [11]. In cases where the average trade is expected to be very large or very small, there will either be a deluge or a dearth of arbitrage opportunities. The steps for trading are as follows:

1. Trader uniformly at random determines a token Y to trade in portfolio

2. Trader draws a number X from a truncated Pareto distribution with probability density function f defined by

$$f(x) = \frac{\alpha L^{\alpha} x^{-\alpha-1}}{1-\left(\frac{L}{H}\right)^{\alpha}} \tag{7}$$

with $\alpha = 0.5$, $L = 1$ and $H = 10000$.

3. Trader swaps X basis points of their holdings in Y on the cheapest to execute pool

The mean of the truncated Pareto distribution is given by

$$\mu = \frac{L^{\alpha}}{1-\left(\frac{L}{H}\right)^{\alpha}} \left(\frac{\alpha}{\alpha-1}\right)\left(\frac{1}{L^{\alpha-1}} - \frac{1}{H^{\alpha-1}}\right) \tag{8}$$

for $\alpha \neq 1$ so plugging in $\alpha = 0.5$, $L = 1$ and $H = 10000$ gives $\mu = 100$ so in expectation the trader exchanges 1% of their holdings in an asset per swap. Using this distribution allows for trades of varying magnitude to be accounted for in the simulation, ensuring that there are interactions with both the CLP and XYK pools.

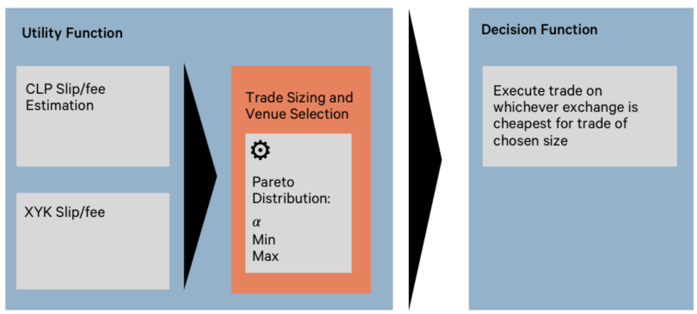

Figure 6: Demand Trader Agent Logic

5 Key Model Parameters

Within our models, we use on-chain data wherever possible to ensure the agent models used in the simulation are as representative as possible of what we expect to see on mainnet. However, it is not a guarantee that what we have seen in the past continues in the future. We still need to make assumptions in the model which will affect the results. We vary these assumptions as widely as possible so as to be sure that results remain valid given a wide variety of conditions on mainnet.

5.1 Starting LP Pool Depths

We initialize the simulation with varying amounts of capital allocated to each exchange. Since exchange liquidity has such a large impact on success, we want to understand what the minimum liquidity THORChain would need to achieve to compete with a larger XYK exchange. In the case where the CLP has much less liquidity than the XYK exchange, it will face difficulties competing with the larger XYK exchange. However as we start to increase the liquidity, measured by the ratio of pool depths between the two exchanges, we can see at what point the CLP exchange starts to be more competitive. This allows us to make sure that our assumption on the relative liquidity between the two exchanges does not create a confounding variable in the simulation results.

5.2 Trade Size

As DEXs have evolved from niche product to major trading venue, there has been a natural increase in trade sizes over time. Moreover, as crypto markets evolve, we expect to see continual changes to multiple properties of empirical trade size distributions. When analyzing fixed-fee exchanges like Uniswap and Balancer, trade size is less important, as the fee is a percentage of total volume. However with THORChain's slip-based fee, trade sizes have a large impact on the total fees generated. In both exchanges, the size of organic trades have a strong affect on arbitrage volume. In our model, we have separated the organic demand and the arbitrage trade volume with the demand trader agent and the arbitrage agent to capture this effect. This allows us to more precisely analyze the revenue split between active market participants in THORChain.

5.3 Trade Volumes

Another important assumption is the trade volume in the simulation. Both the total volume traded and the average size effects the success of each fee model. When trades are small relative to the pool size, THORchain is able to draw trades away from XYK exchanges as the lower slip fee incentivizes these trades.

5.4 Asset Price Trajectories

The price trajectories mentioned in section 3.2 have a large effect on the results of the simulation. These affect the impermanent loss experienced by LPs. We define impermanent loss in 6.2.3 and demonstrate that it is a large factor in LP returns. Unlike fixed-fee XYK exchanges, where the parameters of the price trajectory fully determine the loss, the slip-based fee adds an extra component to the analysis of impermanent loss.

6 Measures of Protocol Success

As we simulate this ecosystem under various different conditions, it naturally begs the question — how do we measure the success of the slip-based fee model? The goal of the slip-based fee is to:

1. Allow LPs to weather drifting price trajectories, which tend to create unfavorable outcomes as shown in 6.2.3.

2. Minimize the impact on trading volume and activity from multiple market participants

6.1 Market Share

There are two measures of an exchange's market share that are used in simulation. The *fee share* is the percentage of fees the total fees that each exchange earns, and the *volume share* is percentage of total volume that trades on each exchange.

6.1.1 Fee Share

The *fee share* for an exchange x is the sum of the fees paid on the exchange, divided by the total fees paid over the lifetime of the simulation.

6.1.2 Volume Share

The *volume share* for an exchange x is the sum of the trade value (*not* including fees paid) on the exchange, divided by the total trade volume in the simulation. This is also measured over the lifetime of the simulation run.

6.2 LP Returns

6.2.1 LP ROI

Liquidity provider ROI is calculated throughout the simulation by taking the current total value of the LP shares and dividing the by the initial value of the assets deposited when the LP first provided liquidity. The value of the LP shares is determined by taking the price of each asset and multiplying it by the amount of that asset in the pool.

$$\text{ROI} = \frac{\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Assets at Time } t}{\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Starting Assets at Time } 0} - 1 \tag{9}$$

6.2.2 LP APY

Since APY takes continuous compounding into account, it is computed by

$$\text{APY} = \left(\frac{\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Assets at Time } t}{\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Starting Assets at Time } 0} \right)^{\frac{\text{Year}}{t}} - 1 \tag{10}$$

6.2.3 LP IL

Ignoring gains from liquidity fees and assuming constant pool share, impermanent loss [12][13] is a term that refers to the change in portfolio value of a LP's pool share relative to simply holding tokens in wallets or centralized exchanges when prices vary from the original staking price. We note that impermanent loss has been studied in traditional finance under the guise of volatility harvesting [14]. From the XYK formula underlying the pool computations if the pool has quantities X and Y of assets x and y respectively then the product $XY = K$ is invariant for some constant K through a transaction (though over time this grows as fees are added). As an example suppose x denotes ETH, y denotes the USD Stablecoin Dai (DAI), p denotes the price of ETH in DAI and assume that the pool is efficiently priced so $Y/X = p$ then originally

$$X = \sqrt{\frac{K}{p}} \quad Y = \sqrt{Kp} \tag{11}$$

Thus the overall pool value V is

$$V = p_x X + p_y Y = p\sqrt{\frac{K}{p}} + 1\sqrt{Kp} = 2\sqrt{Kp} \tag{12}$$

Suppose that the price moves by a factor of $c > 0$ so the new price $p' = cp$. Then the new pool value V' is

$$V' = p'_x X' + p'_y Y' = cp\sqrt{\frac{K}{cp}} + 1\sqrt{Kcp} = 2\sqrt{Kcp} \tag{13}$$

Using the constant pool share assumption, if a staker had instead held the assets instead of staking their portfolio share scaled to the entire pool W would be worth

$$W = p'_x X + p'_y Y = cp\sqrt{\frac{K}{p}} + 1\sqrt{Kp} = (c+1)\sqrt{Kp} \tag{14}$$

Note that $c+1 \geq 2\sqrt{c}$ by the AM-GM inequality so there is always a relative drop in value in this no fee constant pool share context. Some intuition for this comes from

$$\frac{\mathrm{d}X}{\mathrm{d}p} = \frac{-1}{2}\sqrt{\frac{K}{p^3}} < 0 \quad \frac{\mathrm{d}Y}{\mathrm{d}p} = \frac{1}{2}\sqrt{\frac{K}{p}} > 0, \tag{15}$$

so as p increases so does Y while X decreases and as p decreases Y does as well while X increases. More generally, changes to p increase the delta to the weaker performing asset since initial stake and decrease the delta to the stronger asset.

$$\frac{\mathrm{d}^2V}{\mathrm{d}p^2} = \frac{\mathrm{d}}{\mathrm{d}p}\left(\frac{\mathrm{d}V}{\mathrm{d}p}\right) = \frac{\mathrm{d}}{\mathrm{d}p}\left(\sqrt{\frac{K}{p}}\right) = \frac{-1}{2}\sqrt{\frac{K}{p^3}} < 0 \tag{16}$$

shows that V is concave and the portfolio delta is inversely related to price, akin to the gamma of a short option position. Holding the assets has a linear delta to price and the holding value W is tangent to the pool value V at the staking price and from concavity it follows that $W \geq V$. This is seen in Figure 7. We define the *Impermanent Loss* as

$$V' - W = -(c + 1 - 2\sqrt{c})\sqrt{Kp} \tag{17}$$

6.2.4 LP Decisions

Two important factors in determining an LP agent's actions in a time step are the rate of return of fee income r_{LP} as an LP in a pool and the historical rate of return of ETH/DAI $r_{\text{ETH/DAI}}$. The former is given by

$$r_{\text{LP}} = \frac{\text{LP Income}}{\text{LP Asset Value}} \tag{18}$$

where an annualized estimate of an LP's fee income from the pool is given by

$$\text{LP Income} = \frac{\text{Year}}{w\Delta t}\sum_{t=T-w}^{T-1}\frac{\text{Pool Fee Income}_t \cdot \text{Agent LP Shares}}{\text{Total LP Shares}} \tag{19}$$

where Δt is the length of a time step, w denotes the length of the lookback window in time steps and T is the current environment time step.

We determine $r_{\text{ETH/DAI}}$ by annualizing a time weighted exponential moving average (EMA) of the returns of ETH/DAI at every time step over the lookback window. This is given by

$$r_{\text{ETH/DAI}} = (1 + X_t)^{\frac{\text{Year}}{\Delta t}} - 1 \tag{20}$$

for $t = T - 1$ where X_t is the EMA of the returns at time step t defined through the recursion

$$X_t = \begin{cases} \frac{r_t + (w-1)X_{t-1}}{w} & T - w \leq t < T \\ 0 & t < T - w \end{cases} \tag{21}$$

In the simulation we specify the impermanent loss adjusted LP APY $r_{\text{LP adj}}$ as a function of r_{LP} and $r_{\text{ETH/DAI}}$ and setting the price change factor from Equation 17 as $c = 1 + r_{\text{ETH/DAI}}$ we get

$$r_{\text{LP adj}} = \frac{2\sqrt{1 + r_{\text{ETH/DAI}}}}{2 + r_{\text{ETH/DAI}}}(1 + r_{\text{LP}}) - 1 \tag{22}$$

At every time step an LP agent computes $r_{\text{LP adj}}$ and if $r_{\text{LP adj}} > r_f$ they add liquidity to the pools and if $r_{\text{LP adj}} \leq r_f$ then they withdraw their liquidity where r_f is the risk free rate of return set in the simulation as 1%.

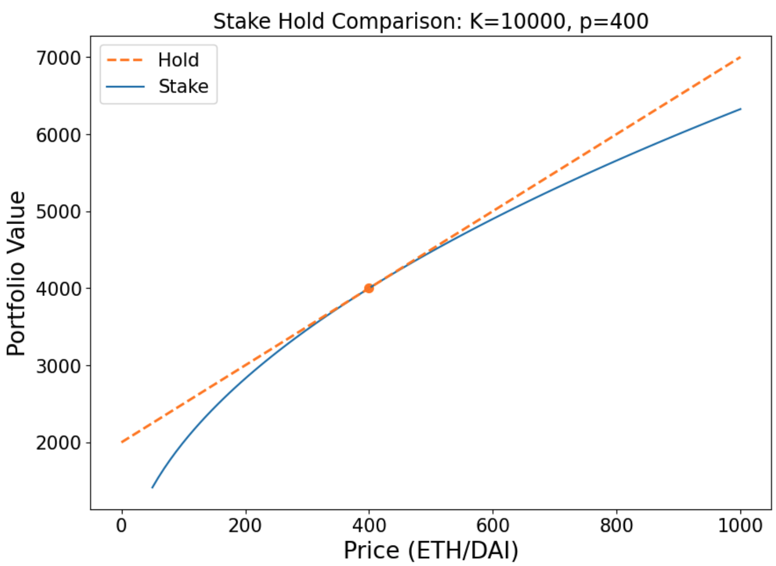

Figure 7: Impermanent Loss

7 Simulation Initialization

7.1 Price Paths

We vary the volatility and drift of the price paths to ensure that they are not skewing our analysis. However, on runs where we vary other parameters, like the pool liquidity, we use a default value for the volatility and drift of the GBM, shown in 1 where volatility and drift are quoted in annualized percentage terms and 0% drift indicates neither positive nor negative bias.

Parameter	Default Value
ETH/USD Initial	400
ETH/USD Drift	0%
ETH/USD Volatility	60%
DAI/USD Initital	1
DAI/USD Drift	0%
DAI/USD Volatility	5%

Table 1: Default price path values

Parameter	Default Value
No. of LP Agents on CLP	10
No. of LP Agents on XYK	10
No. of Arbitrage Agents	10
No. of Demand Trader Agents	10

Table 2: Agents

7.2 Agent Initialization

The number of agents and their individual bankrolls are also key parameters in determining the outcome of the simulation. In particular the demand trader and LP initialized wealth are varied throughout to account for a wide range of scenarios. The default values are shown in 2 and 3. The arbitrage agents have smaller bankrolls since they perform all legs of the arbitrage with their entire bankroll immediately as presented and the largest DAI order size we trained for in our slippage models was $100\,000$. To ensure that arbitrage opportunities are fully realized, arbitrage agents are allowed to trade up to 10 times in a time step.

While the demand trader and LP bankrolls are listed in USD their holdings only ever consist of ETH and DAI with the default values obtained from marking their starting wallet to initial token prices. LPs start with an equal 50/50 notional split of ETH and DAI to allow for efficient liquidity provision whereas demand traders begin with an 80/20 split of ETH and DAI to more accurately reflect the fact that ETH circulating supply is much larger than any stablecoin's, including DAI. As a default the CLP and XYK LPs have the same amount of available capital though we progressively increase the XYK LP wealth to highlight boundaries where added pool depth cancels out some of the advantages of the slip-based fee.

Parameter	Default Value
Arb Agent Bankroll Size	$\$100\,000$
Demand Trader Bankroll Size	$\$20\,000\,000$
LP Agent Bankroll Size	$\$8\,000\,000$

Table 3: Agent Parameters

7.3 Time

The simulation occurs over a period of 6 months with time steps of 1 day. This was chosen to strike a balance between a long enough horizon to account for LP lookback windows of sizably different lengths but short enough to acknowledge the rapidly changing AMM landscape.

Part III

Simulation Results and Analysis

8 Price Trajectories

ETH/USD annualized drift and volatility provide the impetus for arbitrageur volume and introduce impermanent loss (IL) for LPs. Since their variation affect nearly every measure of protocol success, we investigate by running a total of 3200 simulations across a search space of

$$\begin{aligned}&\text{ETH Volatility Annual (\%)} \in \{10, 30, 60, 100, 150, 210, 280, 360\}\\&\text{ETH Drift Annual (\%)} \quad\; \in \{-100, -50, -25, 0, 25, 50, 100, 200\}\end{aligned}$$

The metric we choose to measure is the LP vs HODL APY where HODL refers to simply holding the assets. Similar to 10, this is defined as

$$\text{LP vs HODL APY} = \left(\frac{\displaystyle\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Assets at Time } t}{\displaystyle\sum_{i=1}^{N} \text{Valuation of LP } i\text{'s Starting Assets at Time } t} \right)^{\frac{\text{Year}}{t}} - 1 \tag{23}$$

where the only difference is the valuation of the starting assets is updated to the price trajectories as time progresses thereby providing a comparison between the strategy of an LP to just HODLing.

LPs in both the CLP exchange and the XYK exchange are making a bet that fees collected on the exchange exceed the IL (eqn. 17). Since price paths are the most significant factor determining the IL experienced by LPs, we show how the different price trajectories affect the APY in the XYK and CLP exchanges in Figures 8 and 9.

These figures show the value of becoming an LP — the estimated return of changing your assets into LP shares. For the most part these highlight the general deterioration of returns with increasing drift and volatility as expected with larger IL. One exception common to both plots is the return gradient under 200% ETH drift. As formulated in section 6.2.4 LP agents decide to add assets to the pool if their estimated impermanent loss adjusted return is more than the risk free rate [3]. As such being an LP is in some sense a mixed strategy of selectively providing liquidity and HODLing depending on the empirical estimation of impermanent loss.

Another important detail is the total fee income in these simulations is substantial with an average daily demand volume of $\$10\,000\,000$, giving a sizable incentive to provide liquidity. This is split among the number of active LPs with a secondary effect of pool depth influencing competition for trades with the other DEX. In the case of 200% ETH drift and 10% volatility the impermanent loss isn't sufficient to deter LPing so the LPs still overwhelmingly provide liquidity and do minimal HODLing. This is the worst case for LPs as HODLing is most profitable in this regime but they are still all deciding to maximally split the relatively fixed amount of pool income. As volatility increases the price movements make the LPs more conservative to deploy their capital so the overall collective strategy becomes more balanced between providing liquidity and HODLing. Since there is less pool competition each individual LPs fee share increases while having assets in the pool. As ETH volatility exceeds 200% annualized the costs of impermanent loss are high enough that liquidity provision occurs much more sparingly and LPs largely favor HODLing, driving the LP vs HODL APY closer to 0. In the tradeoff of higher

[3] Note that LPs do not have a directional market opinion based on historical moves, merely an estimate of volatility used to compute empirical impermanent loss so their decision to not LP and thus HODL is assumed only to have the expectation of earning the risk free rate

fee share vs impermanent loss at an individual LP level the sweet spot appears to be between 150% and 200% volatility, though the increased costs of trading for demand traders in reality would likely lead to a decline in fee income.

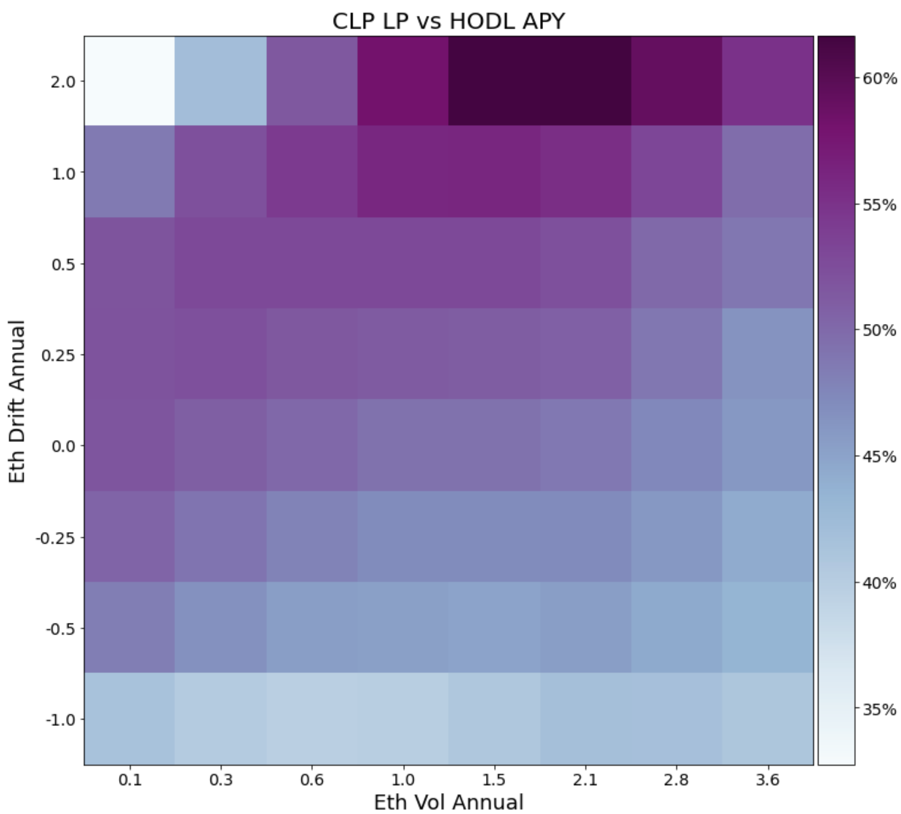

Figure 8: CLP LP Returns, Adjusted for Impermanent Loss (eqn. 25)

In the XYK exchange the LP vs HODL APYs are modest and veer negative in the extreme drift and low volatility cases, stemming from the general lack of protection a fixed fee provides towards the gamma-esque price exposure of LP shares. In contrast, the CLP exchange returns are consistently high across the board. Since the fees taken from arbitrageurs scale with slippage, this generates more income for LPs in times of higher drift and volatility and has the added benefit of deepening the pools more quickly thus increasing price inertia. Impermanent loss doesn't go to zero, though it has a less greatly reduced impact on LP decision making than in the XYK exchange. You'll notice that in 11, the CLP actually doesn't capture a dominant market share of trades since the XYK can be cheaper for traders for larger trades. However since LPs have much better ROI regardless of price path LP capital would be expected to start moving towards the CLP exchange. Once the CLP exchange is able to draw in more capital, the liquidity will improve, as will the prices for traders, followed by market share.

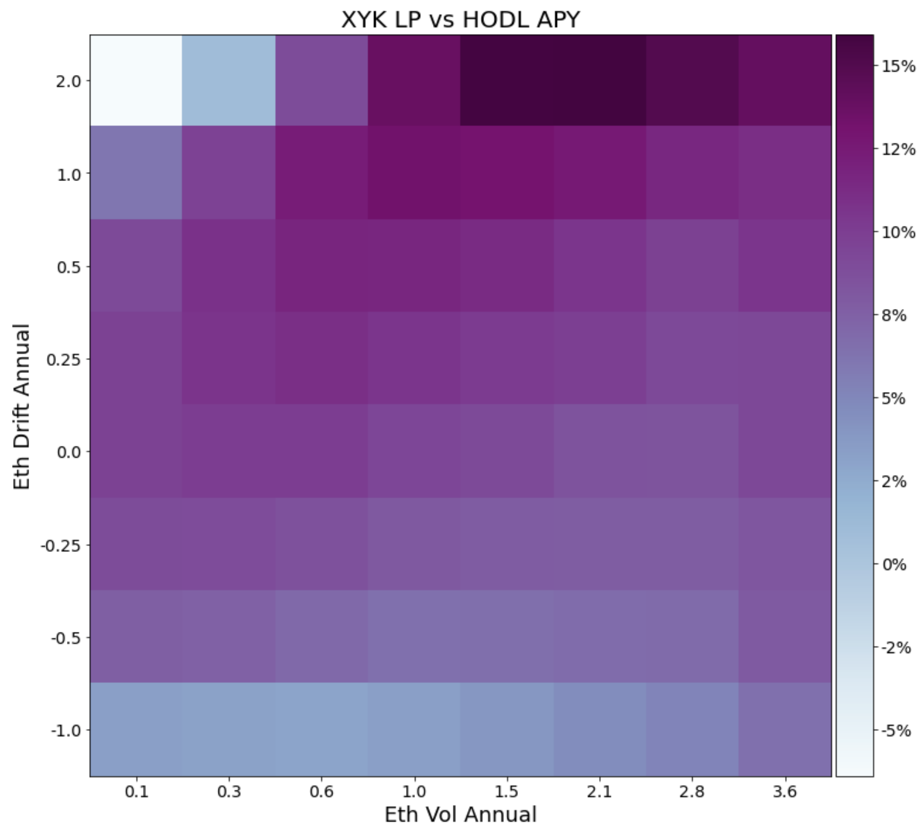

Figure 9: XYK LP Returns, Adjusted for Impermanent Loss (eqn. 25)

Key Takeaway

The slip-based fee improves the expected returns for liquidity providers in most market conditions. Impermanent loss is meaningfully reduced, but not eliminated.

9 Liquidity and Trade Volume

The amount of capital LP agents have available is the primary determinant in the depth of the liquidity pools while the fees generated from demand volume are a key incentive for liquidity provision. To see the effect of these parameters on agent decision making we run a total of 2560 simulations across a search space of

Average Daily Demand Volume (MM USD) $\in \{1, 2, 5, 10, 20, 30, 40, 50\}$
XYK/CLP Depth Ratio $\in \{1, 2, 3, 4, 5, 6, 7, 8\}$

where the depth ratio is defined as

$$\text{XYK/CLP Depth Ratio} = \frac{\text{XYK LP Starting Capital}}{\text{CLP LP Starting Capital}} \tag{24}$$

with starting capital serving as a proxy for pool depth. We focus only on cases where XYK has at least the capital of CLP to give a sense of how much of a capital head start an XYK pool would need to be competitive with a CLP counterpart under market share metrics.

Figures 10 and 11 show fee and volume share generally increasing with demand volume while decreasing with depth ratio. The bottom row in both plots track results for when the DEXs start with same amount of capital where CLP is able to take in 60-90% of the fees while processing 30-50% of the volume. As the initial pool imbalance grows XYK becomes a more attractive trading venue as its deeper pool has lower slippage costs for traders. As expected CLP's market share dwindles progressing towards the top row where XYK has 8 times the capital, CLP collects 10-40% of income and captures 0-30% of the volume. In spite of this, the CLP fee share exceeds the market share across the grid, demonstrating that the CLP model is able to generate fees more effectively than a standard fixed-percentage fee model.

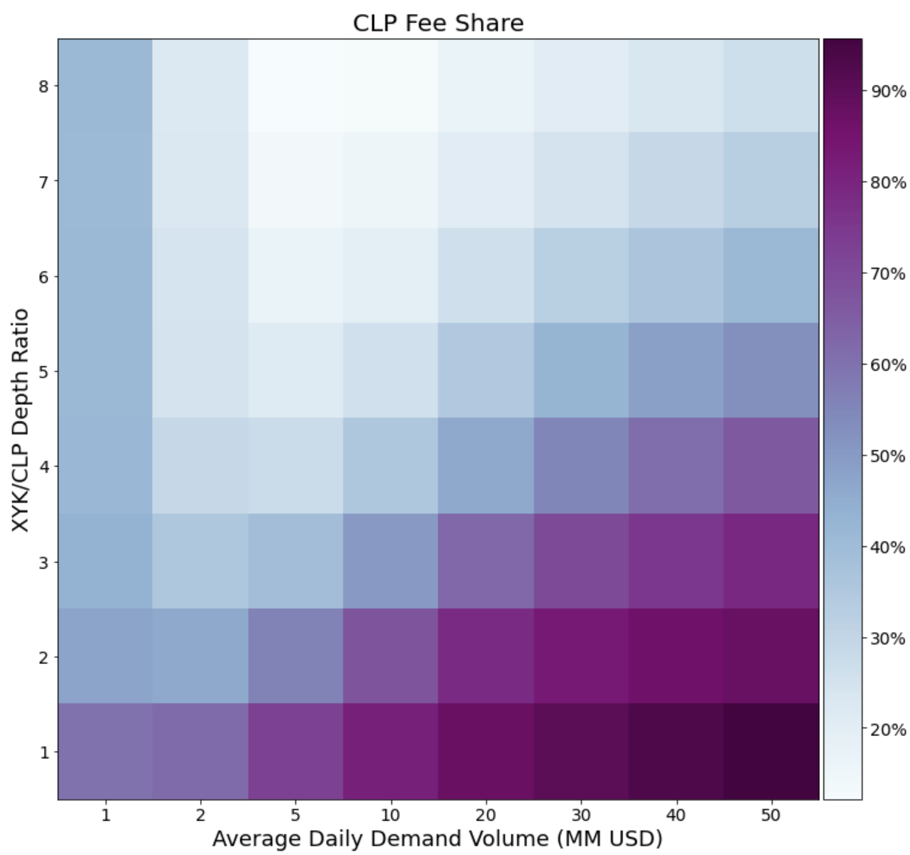

Figure 10: CLP Fee Share vs. Liquidity and Trade Volume

Examining the variation with demand volume, the leftmost column in both plots covers the scenarios where the average daily demand volume is low. Since the trade sizes are small the CLP fee is often lower than the fixed fee of 30 bps so the CLP volume share is consistently above 30% even when the XYK pools are much deeper. In the rightmost column when average daily volume is high CLP is able to take the majority of fees even up to a 5x disadvantage in starting capital while doing as little as 10% of the overall volume. Though trades only occur when the CLP pool prices happen to be attractive enough to warrant the heightened costs, this is still indicative of how well CLP monetizes large trades. As a secondary effect the slip based fee also acts as a variable growth rate for pool depth, allowing CLP to reduce impermanent loss for LPs and slippage for traders more rapidly in periods of highly active trading.

In the middle columns as the volume increases the CLP fee rises and when the XYK pools reach a certain depth advantage they are more attractive in terms of both fee and slippage. Volume share decreases from 10MM daily demand volume onwards when the depth ratio is at least 4, hovering between 5-15%. However when the pools are closer in depth we see another transition where XYK volume share peaks in moderate daily volume conditions of 10-20MM, a sweet spot where the trade sizes aren't large enough for CLP's monetization advantages to offset XYK's lower fees. A similar transition occurs in fee share at a much lower boundary between 2-5MM daily volume and a depth ratio of 3-4 from where the metric increases with higher volume and decreases with capital deficiency. All told, the goal of the CLP is to drive higher fees to the protocol to offset price risk, and these results show that it is likely to succeed in this endeavor.

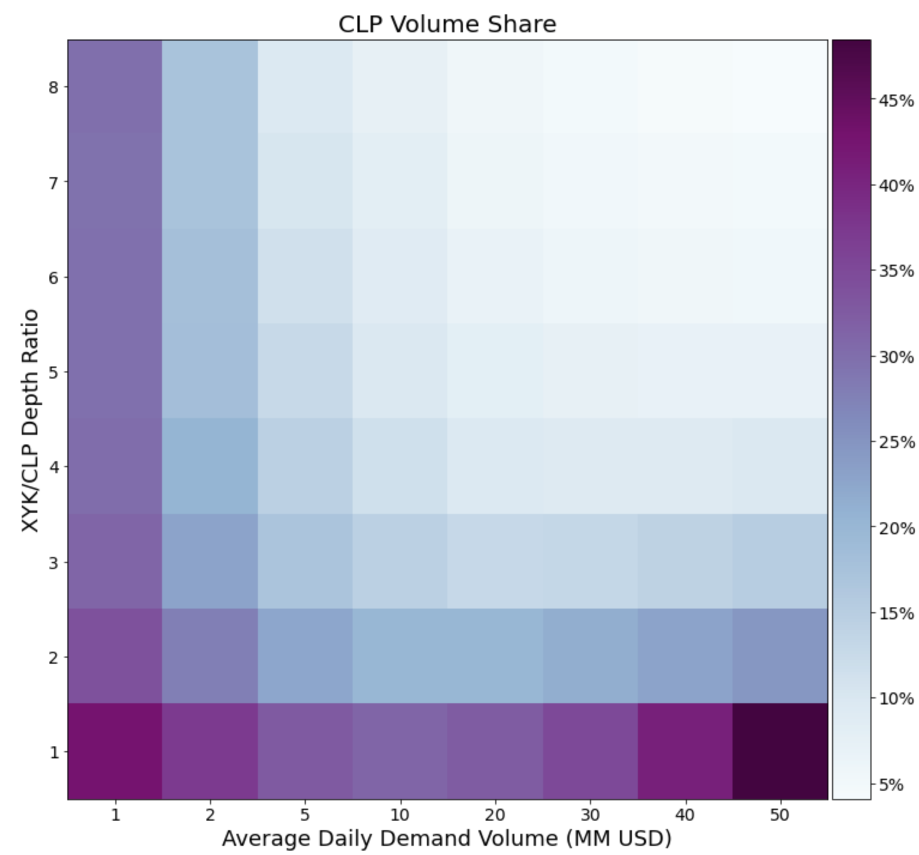

Figure 11: CLP Market Share vs. Liquidity and Trade Volume

Key Takeaway

Returns to LPs appear to be much more consistent in the CLP fee model. Because of this, the liquidity on THORChain should improve to the point where it offsets the fee for larger and larger trades. If THORChain is able to attract sufficient liquidity, organic demand will favor trading on THORChain.

One other factor that affects the trading results of the simulation is how closely the price on each exchange tracks centralized exchange prices. We leave a detailed investigation of this for future analysis, but have included some preliminary results in Appendix D.

10 Results

As seen from the heatmaps in the previous sections, **the slip-based fee improves the expected returns for liquidity providers in most market conditions.** The CLP model does well when there is high volatility between trading pairs, and is able to withstand substantial price drift. Almost all price movement that is not mean-reverting is bad for a standard XYK exchange. However there are many price trajectories where THORChain LPs are still able to make a profit despite lack of mean reversion. **Impermanent loss is meaningfully reduced, but not eliminated.**

The slip-based fee does disadvantage THORChain in a highly competitive market for trading volume, as large trades will be cheaper with a fixed fee. Prices on the exchange still remain competitive for smaller trades, with arbitrageurs able to keep prices close to the values on other exchanges. Arbitrage profits are better in the XYK exchange, but that's expected given the slip-based fee and there is still enough financial incentive for arbitrageurs to keep prices in sync. As returns to LP providers appear to be much stronger, the liquidity on THORChain should improve to the point where it offsets the fees for progressively larger trades. **If THORChain is able to attract sufficient liquidity, organic demand will favor trading on THORChain.**

Part IV
Appendix

A Background on Agent-Based Simulation

The main tool that we use to analyze THORChain's protocol is agent-based simulation (ABS). ABS has been used in a variety of contexts in quantitative finance, including to estimate censorship in cryptocurrency protocols [15], detect fraudulent trading activity in CFTC exchanges [16], and in stress testing frameworks from the European Central Bank [17, 18] and the Federal Reserve [19, 20]. These models can provide invaluable information on the behavior of complex systems. This has made ABS widely used in industries such as algorithmic trading and self-driving car deployment.

B Gauntlet Simulation Environment

The Gauntlet platform, which was used for all simulations and results in this report, provides a modular, generic ABS interface for running simulations directly against Ethereum smart contracts as well as Python representations of contracts. Here we use the latter as the THORChain is not built on Ethereum. In this system, the agent models are specified via a Python domain-specific language (DSL), akin to Facebook's PyTorch[21]. Agents can also interact with non-blockchain modules, such as historical or synthetic market data and/or other off-chain systems. The DSL hides the blockchain-level details from the analyst, allowing the end-user to develop strategies that can migrate from one smart contract to another, should they have similar interfaces. Most of the platform's design is inspired by similar platforms in algorithmic trading that allow for quantitative researchers to develop strategies that execute over multiple exchanges (with varying order books, wire protocols, slippage models, etc.) without having to know these low-level details. Moreover, the non-blockchain portions of the simulation are analogous to trading back-testing environments,[22] so that agents are interacting with realistic order books and financial data.

C Python Contracts

There are several contracts in the simulation environment in addition to our agent based simulation platform:

1. XYK Contract: An implementation of a traditional constant product market maker with a fixed fee on the input token which serves as a decentralized liquidity pool alternate to THORChain.
2. CLP Contract: An implementation of a constant product market maker with a slip based fee on the output token.
3. Exchange Contract: A centralized venue for agents to buy and sell liquid tokens such as ETH and DAI through a traditional order book. Simulates the price impact effects of trades by updating the token prices according to impact model parameters. Acts as an external price feed.

D Price Inertia

If arbitrageurs are not able to keep prices in line with a primary venue, in this case the CEX market, the DEX risks becoming non-competitive. Without fees, there is a clear path to price parity as discussed in [2]. However with the introduction fees, the DEX price will at best track within an error bounds determined by those fees:

$$(1-\tau)m_p \leq m_{\text{DEX}} \leq (1+\tau)m_p.$$

where τ is the fee, m_{DEX} is the price on the DEX and m_p is the market price.

In the XYK exchange, τ is simply 30 basis points, but for the CLP it is determined by the equations given in §3.1.3, which vary with trade size. In theory since arbitrageurs can execute infinitesimally small trades that incur epsilon slip, prices in a CLP should track within a smaller range than on the XYK exchange (though gas costs and network fees impose a limit to how closely). In the simulation minimum arbitrage trade sizes add slippage costs that widen error bounds and we measure mean error as

$$\text{DEX CEX Mean Error} = \frac{1}{T}\sum_{t=1}^{T} \frac{\text{DEX Price at time step } t - \text{CEX Price at time step } t}{\text{CEX Price at time step } t} \tag{25}$$

Both DEXes had reasonable average tracking errors for the CEX prices as seen in Figures 12 and 13. An interesting distinction is that the CLP tracking error follows the direction of the drift which implies that its price is lagging the CEX market. This is expected behavior as a secondary market as defined in the simulation. CLP also tends to track higher than the centralized exchange price as volatility increases. This is in part because the slip based fee enables scalable price corrections so deviations of large magnitude will still be corrected to within arbitrage boundaries. As such the main factor in deciding error direction is then the proportion of up vs down moves in the price trajectory. Recalling that the probability of a price drop in a time step is given by

$$\mathbb{P}\left[Z \leq \frac{-(\mu - \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}\right]$$

where $Z \sim \mathcal{N}(0,1)$ the frequency of down moves increases with volatility as long as $\mu > \frac{-\sigma^2}{2}$.

XYK's mean error follows CLP's closely in the high drift low vol case and tracks fairly symmetrically when volatility is between 50 and 150%. In this range arbitrage costs are still relatively low given the fixed fee so correcting moves in either direction to within arbitrage bounds is straightforward. When volatility gets higher XYK tracks lower than the CEX market. This is due to a combination of LPs suffering greater impermanent loss and thereby leaving the XYK pools shallower and the nuance that arbitrage agent bankrolls are fixed in USD but ETH price up and down moves are symmetric in % terms so it takes more capital to correct price jumps vs price crashes. In a more extreme formulation the lowest the ETH price can be is 0 but the highest it can go is infinite and exhaustively trading the former is possible but the latter is not. Thus large up moves tend to go uncorrected for longer and these are more prevalent with higher volatility.

From these plots, CLP exhibits more price inertia which is useful in deterring sandwich and flash loan based attacks but can lead to consistent lags in price jump events such as breaking news or strong directional trends like broad risk off sentiment. This does raise the possibility of price exploits but in general the error should converge towards 0 with deeper pools and increased frequency of smaller arbs.

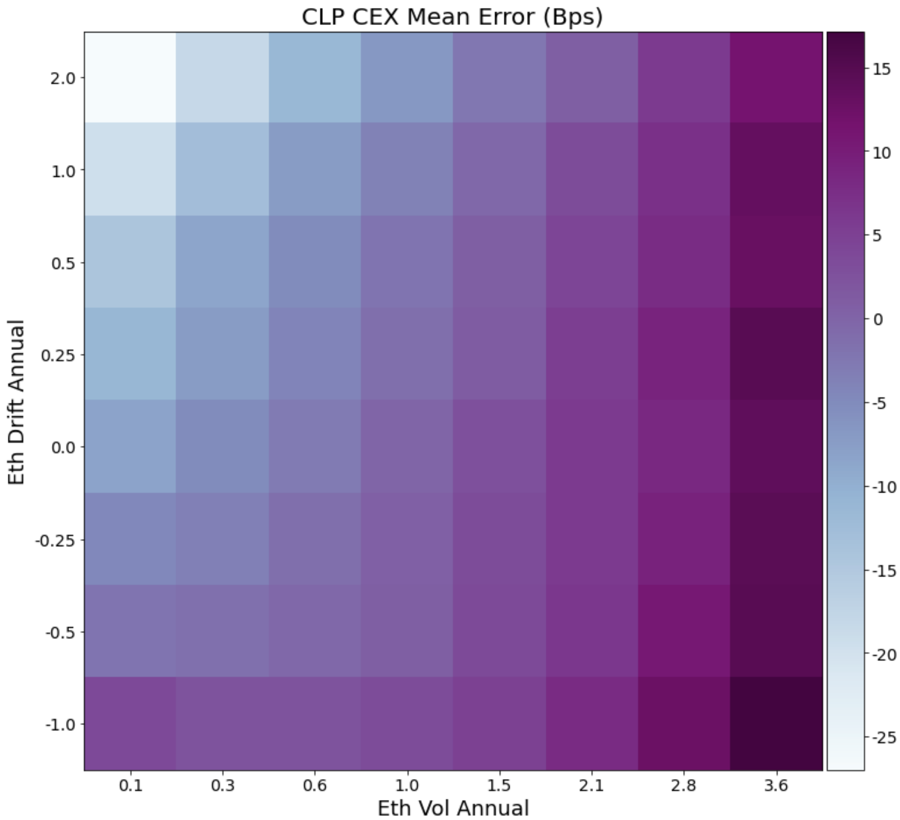

Figure 12: CLP Mean Tracking Error from CEX

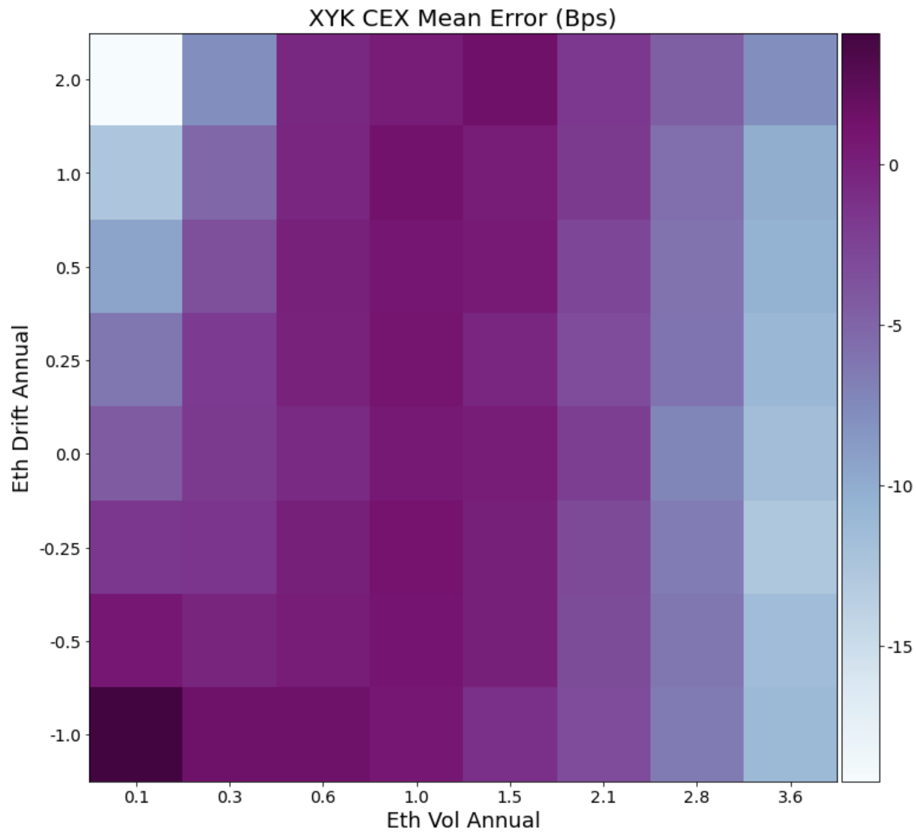

Figure 13: XYK Mean Tracking Error from CEX

Since our demand trading agents take this ”quoted” price into consideration (the price implied by the pool sizes on the DEX), these variations in price are considered in the simulation, but we leave a more in depth investigation of this to a future analysis. To investigate this in detail, it would make sense to look at shorter time scale simulations than the time parameters we chose in 7.3.

E References

[1] Ariah Klages-Mundt and Andreea Minca. (in) stability for the blockchain: Deleveraging spirals and stablecoin attacks. *arXiv preprint arXiv:1906.02152*, 2019.

[2] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. *arXiv preprint arXiv:1911.03380*, 2019.

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

[4] Mika Honkasalo. Uniswap’s monthly trade volume exceeded coinbase’s in september, Oct 2020.

[5] Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, to appear.

[6] Thorchain. Thorchain documentation. `https://docs.thorchain.org/`.

[7] Victor Xu, Hsien-Tang Kao. Thorchain protocol assessment. `https://gauntlet.network/reports/thorchain`, June 2020.

[8] Hsien-Tang Kao, Tarun Chitra, Rei Chiang, and John Morrow. An analysis of the market risk to participants in the compound protocol, 2020.

[9] Uniswap. Uniswap v2 documentation. `https://uniswap.org/docs/v2/protocol-overview/how-uniswap-works/`, 2020.

[10] Synthetix Foundation. Synthetix documentation. `https://docs.synthetix.io/litepaper/#synthetixexchange`, 2020.

[11] Seungki Min, Costis Maglaras, and Ciamac C Moallemi. Cross-sectional variation of intraday liquidity, cross-impact, and their effect on portfolio execution. *Columbia Business School Research Paper*, (19-4), 2018.

[12] Uniswap: A good deal for liquidity providers? `https://medium.com/@pintail/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2`, January 2019.

[13] Understanding uniswap returns. `https://medium.com/@pintail/understanding-uniswap-returns-cc593f3499ef`, February 2019.

[14] Jan Hendrik Witte. Volatility harvesting: Extracting return from randomness. Preprint, 2015. Available at `https://arxiv.org/abs/1508.05241`.

[15] T. Chitra, M. Quaintance, S. Haber, and W. Martino. Agent-based simulations of blockchain protocols illustrated via kadena's chainweb. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pages 386–395, June 2019.

[16] Steve Yang, Mark Paddrik, Roy Hayes, Andrew Todd, Andrei Kirilenko, Peter Beling, and William Scherer. Behavior based learning in identifying high frequency trading strategies. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*, pages 1–8. IEEE, 2012.

[17] Grzegorz Halaj. Agent-based model of system-wide implications of funding risk. 2018.

[18] Anqi Liu, Mark Paddrik, Steve Y Yang, and Xingjia Zhang. Interbank contagion: An agent-based model approach to endogenously formed networks. *Journal of Banking & Finance*, 2017.

[19] John Geanakoplos, Robert Axtell, J Doyne Farmer, Peter Howitt, Benjamin Conlee, Jonathan Goldstein, Matthew Hendrey, Nathan M Palmer, and Chun-Yi Yang. Getting at systemic risk via an agent-based model of the housing market. *American Economic Review*, 102(3):53–58, 2012.

[20] Richard Bookstaber, Mark Paddrik, and Brian Tivnan. An agent-based model for financial vulnerability. *Journal of Economic Interaction and Coordination*, 13(2):433–466, 2018.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[22] Peter Nystrup, Stephen Boyd, Erik Lindström, and Henrik Madsen. Multi-period portfolio selection with drawdown control. *Annals of Operations Research*, 282(1-2):245–271, 2019.

F Authors

Victor Xu is a Data Scientist at Gauntlet. Victor worked as a quantitative trader at Flow Traders. Prior to Flow Traders, Victor spent time at Citadel and Jane Street Capital. He studied Mathematics and Computer Science at Princeton University.

John Morrow is Gauntlet's COO. Prior to Gauntlet, he worked on product and data engineering at Gigster, Microsoft, and Goldman Sachs. He also studied at Cornell where he received a B.S. in Applied Physics and Computer Science.

G Acknowledgements

The authors would like to thank John-Paul Thorbjornsen and Chad Barraford for helpful discussions. We would also like to thank the other members of Gauntlet team, in particular Rei Chiang and Hsien-Tang Kao for simulation design and best practices, Tarun Chitra for guidance and clarity, and Jonathan Reem for help with the simulation infrastructure.