Zero
PEM-AMM : Positive Externality Maximizing Automated Market-Maker

Ulysse Forest
ulysse.forest@epfl.ch

May 26, 2024

Abstract

Automated Market-Makers is a new type of market structure computationally efficient, where users are able to swap one asset for another without a central entity overseeing the trading activity. We introduce Zero Protocol, an AMM eliminating the price-impact of transactions and making liquidity-providing more attractive, while maximizing positive externalities generated by traders. Thus reducing the share of volume handled by *intentional* arbitrageurs.

1 Introduction

Zero Protocol is a new type of Automated Market Maker maximizing positive externalities, mainly those generated by traders performing swaps. One of the main characteristics of the AMM is that all transactions are executed at an Oracle rate, which is independent from the quantities of tokens present in the Pool. The price-impact of transactions is therefore reduced to zero. The real Pool rate, function of the quantities of tokens in the Pool, as in classic AMM models, is now "virtual" and used to measure certain metrics and avoid the risk of penuries.

We introduce a new measure: AVD (Arbitrage Value Delta), the only one directly impacting Liquidity Providers. Unlike conventional AMM models where Liquidity Providers suffered from Impermanent Loss and LVR and a long MEV Attack List (this was also the case for Traders)[1]; AVD is significantly lower than the losses listed above. More importantly, it's not necessarily a loss. Indeed, we show it tends to follow a Geometric Brownian motion and the and the PnL distribution is symmetric between positive and negative. Furthermore, it is possible, using forecasting algorithms, to optimize certain parameters of the Pool to make the AVD positive for LPs.

2 Basics

Let's start by defining the key concepts necessary to understand the model. A Pool consists of the quantities X_t and Z_t of different tokens. We then define the Pool's "virtual" rate,

$$S_{\text{pool}} = \frac{Z_t}{X_t}$$

One of the main features of the AMM is that all transactions are executed at a certain *low-latency* Oracle price [1]: $\Gamma(t)$, we then define Δ,

$$\Delta = \frac{\Gamma_t - S_{\text{pool}}}{\Gamma_t} \cdot 100$$

As transactions change the quantities of tokens present in the Pool, it is quite natural to define 2 types of transactions: α and β. Transactions α generate a positive externality on the Pool, while transactions β generate a negative externality on the Pool. To construct this model, we directly reasoned about the PnL of the Pool, considered as a set of Liquidity Providers. If we take a 50/50 Hodl scenario as a benchmark and consider that LPs do not earn transaction fees, then the Pool's PnL is impacted only by the AVD (Arbitrage Value Delta), given as

$$AVD_t = \Gamma_t(X_t - X_{t_0}) + (Z_t - Z_{t_0})$$

taking its values in $\mathbb{R}$ where X_{t_0} and Z_{t_0} are the initial quantities at the time of the deposit by actors assumed to be *passive*. We then define the positive externality of a type α transaction occurring at date t_1: The AVD_t increases after this transaction, and the negative externality of a type β transaction: The AVD_t decreases after this transaction. These externalities are not taken into account when pricing transaction costs but have a direct impact on the returns of LPs. The type of externality of a transaction is determined by its impact on the AVD_t in the short term up to a time $T > t_1$. This moment is

[1] https://chain.link/data-streams

defined as the date when we assume that Δ has become optimal again.

In traditional AMMs models, the metrics that impacted LPs, other than transaction fees, were inevitably losses (IL, LVR) and, additionally, significant ones. Moreover, traders suffered from a significant price impact on their transactions[2]. It is therefore possible to conclude that these models were absolutely not the best option, as they penalized uninformed active and passive actors [3] and favored price manipulation and MEV attacks.

Now that we have this in mind, we can now address the topic: How is it possible to construct an AMM where the LPs' returns (excluding transaction fees) are potentially positive while ensuring zero price-impact transactions for traders?

Indeed, in previous models, blockchain technology was used both to secure transactions in a completely decentralized manner and to house the characteristic AMM algorithms, often functions of the quantities of tokens present in the Pool that did not take any external factors into account and could never be optimized numerically because they were completely deterministic. Our approach is different: Zero uses the blockchain to ensure completely decentralized exchanges and liquidity and to make visible to everyone the parameters that determine the behavior of the Pool, which everyone is free to consult before their transaction for complete transparency. Indeed, the essential aspect of decentralization is preserved while opening up optimization possibilities for an attractive environment for Traders and Liquidity Providers, and retaining a key advantage of AMMs: A transaction, to be executed, does not require two active actors to agree on a price, but rather a passive actor (LP) allows the active actor to perform swaps at any time.

3 Public parameters

The model is designed with a set of parameters, visible on-chain to everyone. Zero-Labs is tasked with optimizing these parameters. Regular potential changes are periodically proposed to governance at a set frequency. All computations are conducted off-chain due to cost and performance considerations. Governance may agree on certain off-chain algorithms updating parameters at high frequency, to deliver optimal experience for traders and Liquidity-Providers. These parameters serve as the public constants of the AMM, analogous to the traditional AMMs model's $f(x, y) = k$. Here are the most relevant ones,

- `Amplitude` — The percentage of the Pool Notional Value that can be exchanged on each-block of the determined Blockchain. Its optimized value takes into account *future* predictions made by off-chain algorithms, as well as *historical* data (for instance: Increase its value if a small number of transactions have been carried out before).

- `OptimalDelta` — The Delta of the pool assumed to be optimal. This target is reached thanks to the positive externality generated by traders performings swaps, who unknowingly rebalance the pool, and by the conditional remuneration of α transactions. The risk-neutral value of this parameter is 0.

- `BlockInterval` — Determines when α transactions become remunerated. Depends on the probabilty distribution of Block times in specified Blockchain. The goal here is to maximize : i) The probability of unintentional Arbitrage to happen. ii) The defined *utility* of the AMM. iii) The probability of Delta staying close to `OptimalDelta`. While ensuring that the AVD per unit of time remains *almost surely* bounded.

- `TransactionFees` — The fees paid by traders. This parameter can be significantly lower in our model than in traditional AMM models.

- `LpWithdrawalFees` — The fees paid by Liquidity Providers withdrawing their liquidity from the pool. Ensuring they are passive actors.

- `IntentionalArbReward` — When the externality generated by some traders performing swaps is not enough, it is neccessary for the arbitrage to become remunerated. Noting, that the PnL of this Arbitrage could be the same as the CEX-DEX arbitrage, while being rewarded at a fixed rate. Ensuring predictability and a near zero Market-risk for arbitrageurs. This parameter should be lower than `TransactionFees`. Intuitively, the lower the Remunerated Arbitrage Notional is, the lower is the cost for LPs.

- `OptimizationFees` — The share of the fees paid to the protocol responsible for the customer interface and for optimising the Pool's parameters. They will also be used to fund research. We estimate this parameter to be less than 1 bips.

4 Technical mechanisms

In this section, we address the innovative mechanisms, built on top of the parametrization described above, used by the model and primarily designed to maximize the positive externalities of trading and Liquidity-Providing activities on the AMM. Indeed, we modify a few key elements that do not impact the experience of traders or Liquidity Providers, but significantly improve the risk profile of Liquidity Providers, while still allowing traders to perform transactions with zero price impact and reduced fees.

4.1 Trading

We assume that the main objective of traders is to execute transactions as quickly as possible, at the best price, and with minimal slippage. Indeed, in traditional AMM models, slippage depends mainly on a list of elements: i) The delay for the entire transaction to be executed — primarily depends on the TTF of the blockchain where the trade is executed, but not only: Uniswap X ii) The price impact of the transaction — Inevitable but measurable on models where pricing is a function of the quantities of tokens in the Pool. Inevitable also (for a certain notional) and difficult to measure on Order-Book models, where traders also pay a spread to remunerate active market makers, often centralized entities. iii) Front-Running and Back-Running, which can be considered a certain type of MEV (Miner Extractable Value) — Their magnitude depends on the AMM considered, but it is quite reasonable to assert that they are absolutely present in all decentralized exchanges in various forms: JIT, Sandwich Attack, Adversarial Arbitrage, Flash Loan attack, etc. (This list is not exhaustive) [1].

By taking a rate independent of quantities and assumed to be immutable as a reference for exchanges, most of these attacks are eliminated. As for the transaction time, it is intrinsically linked to the blockchain or distributed ledger in general. Reducing it means using faster networks than Ethereum, which the user, after consideration, is free to do, as Zero would be present on major distributed ledger technologies.

On Zero, traders can place different types of orders: i) Market Order — Executed in its entirety [2] as quickly as possible at the Oracle rate. On Zero, regardless of its size, it will have no price impact. ii) Limit Order — Executed when the Oracle rate reaches the specified limit price. It can also set a bound on the tolerated price change. A default value is dynamically suggested by Zero-Labs based on certain metrics (volatility, volume, etc.) and the network on which the order is executed. An execution probability is then displayed. This is the recommended order type as it allows for protected execution. In traditional AMM models, which suffered from MEV attacks as detailed above, this type of order provided protection. iii) Stop Order — Executed as a market order when the stop price is reached. In short, executing an order on Zero means benefiting from an execution price based on a consensus of centralized exchanges, optimally reflecting the real price of the asset; all without price impact, at reduced fees, and completely decentralized; the model could thus be likened to a "Dark Pool" accessible to all.

It is necessary to address the case where the maximum `Amplitude` per block is reached. First of all, we show in section 6.Analysis that this scenario is extremely unlikely to occur. By way of comparison, it would require approximately, with equal liquidity, 15 times the trading volume of Uniswap [3]. However, we then introduce a fairly trivial "transaction-queue" mechanism where limit and stop orders have priority. Indeed, if a transaction cannot be executed, it is placed, with a specified index, in the list of pending transactions. Pending transactions then have priority over new transactions. There is also a waiting list for limit orders and stop orders for each price level if it is estimated that the amplitude will necessarily be exceeded by summing the estimated notional of these orders. As detailed in the Analysis section, these scenarios are extremely unlikely to occur. For example, under normal circumstances, transactions via UniswapX take a long time to be completed by centralized market makers [4].

Note — It would be possible not to consider a maximum `Amplitude` per given time period, as the traditional AMM models did, but the potential losses related to LVR [4] and IL [5], which were directly linked to the total notional of transactions over a given time scale, were then unlimited (infinite price manipulation was also possible). The objective here is to limit the dispersion of AVD loss for Liquidity Providers and thus improve their risk profile, so that price manipulation is no longer possible.

Now, let's see how it is possible to maximize a certain externality generated by traders performing transactions — We will act on an element that does not impact transaction costs but has a measured impact on the Pool: the token with which transaction fees are paid.

Clearly, we want this token to be determined based on the Pool's Δ and `OptimalDelta`, and chosen in such a way that the externality is *positive*.

Finally, a possible analogy with an Order Book would be to say: It is possible for a trader's order to be intentionally filled by another trader. We have thus reviewed the new mechanisms that have a beneficial impact on the traders' experience that Zero adopts on top of those of traditional AMMs. We will now focus on the mechanisms introduced related to Liquidity Providing.

4.2 Liquidity-Providing

Recent types of AMMs [6] have developed models allowing Liquidity Providers to be active and competitive among themselves by providing *concentrated* liquidity. But what is the point of an AMM if it requires two active actors (LPs and arbitrageurs) simultaneously to allow transactions to take place? This would be a much less efficient structure than an Order Book, which requires only one active actor (Market Makers) for transactions to occur. Moreover, a study shows [7] that purely passive LPs generally lose money. We believe that the main role and the very essence of AMMs is not to increase the number of necessary active actors, but rather to reduce

[2] If the available `Amplitude` allows

[3] https://info.uniswap.org/

[4] https://www.gauntlet.xyz/resources/uniswap-price-execution-analysis

it. Thus, on Zero, Liquidity Providers are intended to be passive actors and not to seek competitiveness among themselves. The role of active actors is assigned solely to arbitrageurs. We also recall that one of the objectives of our model is to reduce the share of arbitrage in the total volume by making it available only at certain times (`BlockInterval`) and by using the potentially positive externalities generated by traders.

We consider that a Liquidity Provider can have 3 Benchmarks: i) the 50/50 Hodl scenario. ii) The initial value of their Portfolio in terms of Z_t, the stable asset of the Pool (for example USDC, USDT, BTC). iii) The initial value of their portfolio in USD. For ii) and iii) we then integrate a feature that allows the Liquidity Provider to hedge against market risk (except AVD) via a collateralized loan on @AaveV3 [5]. This feature aims to attract new Liquidity Providers who wish to obtain a yield based on transaction fees in a fully decentralized manner, and also to improve the experience of existing Liquidity Providers by ensuring a purely passive role, drastically reducing the volatility of their portfolios. For i) we recall that the Liquidity Provider is only exposed to AVD_t.

The main feature offered to Liquidity Providers is the constant optimization of Pool parameters (Section 3) so that they benefit from a model that is high-frequency adapted to real-time market conditions, rather than their capital depending on a function (e.g., f(x,y)=k) that is completely independent of external factors.

Transaction fees are directly added to the pool; there is no separate fee reserve. The auto-compounding is intended to directly contribute to gradually reducing the volatility of the AVD. Additionally, this prevents a potential fee reserve from being insufficient to remunerate the arbitrage. The PnL in fees of an LP could then theoretically be negative. However, the parameters are chosen so that this almost surely never happens. We show in the analysis section that even with remuneration equal to transaction fees, such a scenario is extremely rare (see Figure 1), or even nonexistent. We recall that the nature of the payment of these fees is determined in the token that maximizes the externality of the transaction: to ensure that the latter is positive and, if possible, maximal. The choice of an auto-compounding mechanism also allows increasing the potential Amplitude, thus generating more and more zero Price-Impact transactions.

In summary, on Zero, a Liquidity Provider is intended to be a passive actor, able to choose directly on the interface whether or not to protect themselves from market risk. They will benefit from continuous optimization of the Pool parameters by Zero-Labs to ensure they experience competitive and less volatile returns compared to other traditional AMM models. We will detail further what a Liquidity Provider can expect in the next section.

[5]https://app.aave.com

5 Intuitive simplified analysis

We propose preliminary but generalizable simulation results conducted on historical data with several sets of static parameters to make empirical observations and justify the feasibility of the model. We also share the assumptions made regarding market conditions and discuss them.

5.1 Assumptions

H1: Buy and Sell Volumes are equals

H2: The Block time is assumed to be 12 seconds.

H3: The total notional value exchanged on each-block follows a uniform distribution in the range of 0 to `Amplitude`.

H4: When `BlockInterval` is reached, we assume the necessary remunerated arbitrage is conducted in a single block.

H5: The Pool is composed of one global Liquidity-Provider, who keeps his liquidity in the Pool during the simulation.

Discussion — H2: Based on the distribution of block times on Ethereum (ref), this assumption simplifies the model by fixing a set time interval for transactions. H1 and H3: Simplify market dynamics in a fairly realistic manner, in the sense that prices are not directly impacted by transactions on Zero. H4: This assumption is completely realistic, especially in an environment where the presence of MEV actors (MEV bots) is considerable. H5: Allows focusing solely on the Pool's yield.

5.2 Simulation and Results

As the *Oracle Rate*, we consider the second tick data of **BTCUSDT** fetched from the Binance Exchange[6]. We define the initial state t_0 of the Pool as follow : $X_{t_0} = 1$ and $Z_{t_0} = X_{t_0} \cdot \Gamma_{t_0}$. We run the simulations from the 01/01/2024 to 02/01/2024, as the model can be considered ergodic, since the data, at this frequency, tends to repeat itself.

`Parameters`	Values
`Amplitude`	1, 2, 4
`OptimalDelta`	0, 0, 0
`BlockInterval`	3, 3, 3
`TransactionFees`	0.0001, 0.0001, 0.0001
`IntentionalArbReward`	0.0001, 0.0001, 0.0001

Table 1: Static parametrization.

[6]https://data.binance.vision/data

These static values were determined empirically and are certainly not optimal values, but they are still relevant given the assumptions we have made. We can first plot the Value of the Liquidity Provider Portfolio, as shown in Figure **??** in the appendix.

For traders, the conditions remain unchanged: no price impact for transactions, and they pay a fee of 1 bps per transaction. Therefore, we focus our analysis on the Portfolio of a Liquidity Provider at Zero. Considering such a benchmark is quite reasonable in the sense that it is entirely possible, as detailed in the Liquidity-Providing section, to hedge a 50/50 portfolio very simply, even if some LPs prefer to keep some market exposure. In our scenario, let's then analyze what the magnitude of the AVD on the Portfolio of a Liquidity Provider would be with the parameters specified in red; this is visible in Figure 1.

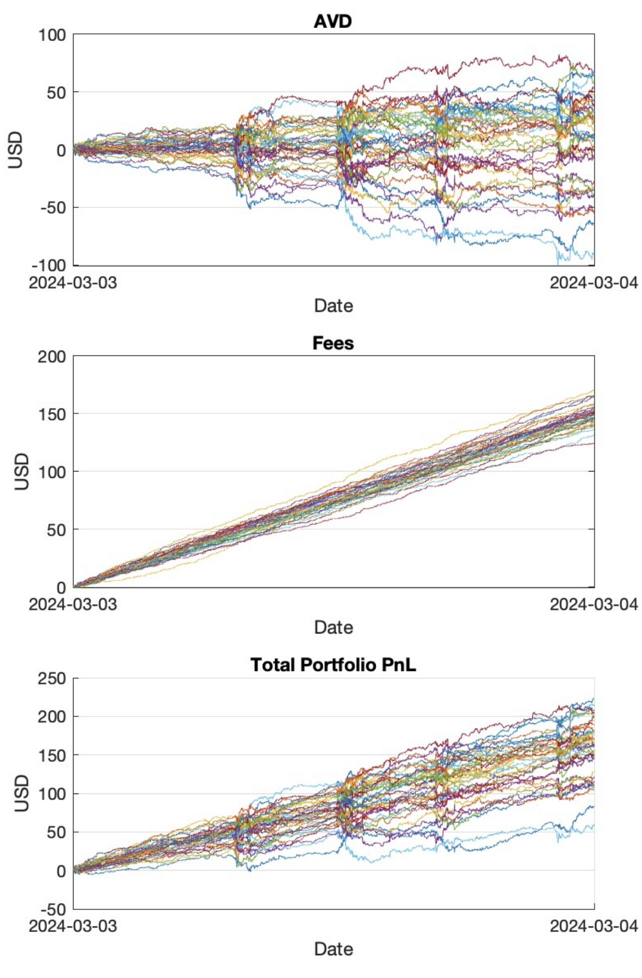

Figure 1: Simualtion of red static parametrization from 2024/03/03 to 2024/03/04 with $X_{t_0} = 1$ and $Z_{t_0} = X_{t_0} \cdot \Gamma_{t_0}$. Benchmark : 50/50 Hodl.

The results obtained with (blue) and without (red) taking transaction fees into account are presented in Table 2. Given that the returns are never negative, the Sortino ratio tends to infinity. This can be interpreted by the fact that, on average, the transaction fees paid by traders are higher than the AVD loss over each `BlockInterval` time interval. Recall that the AVD can also be positive. For comparison, the returns of a Liquidity Provider on traditional AMM models are significantly less attractive[5], in addition to being subject to many MEV attack risks[3].

`Metric`	Values in USD
`meanAVD`	-4.5678
`stdAVD`	52.1947
`meanTotalPnL`	145.6932
`stdTotalPnL`	45.9381
`maxDD`	+52.5894
`SharpeRatio`	2.85947
`SortinoRatio`	∞

Table 2: Red Static Parametrization results.

6 Case study: `Amplitude`

Given the results presented above, one might wonder why we limit the amplitude. Since, for the PnL of Liquidity Providers, although this increases dispersion, their expected value remains the same. The main problems with this reasoning are: i) It increases dispersion, which decreases the Sharpe ratio. ii) Most importantly — The notional required to perform the arbitrage would be much higher, which can make it difficult to achieve. Additionally, it is more exposed to market variations during `BlockInterval`, let's explore in detail why this is the case.

Suppose that the `OptimalDelta` is zero. Let $\Gamma(t^{\text{arb}})$ be the oracle rate at the time when arbitrage occurs, and $S_{\text{pool}}(t^{\text{arb}})$ be the virtual rate of the Pool at the time when arbitrage occurs (thus directly influenced by the transactions taking place in the Pool). We recall that this concerns intentional arbitrage during the `BlockInterval` phases. Then we can express $\delta(X)$ and $\delta(Z)$, the quantities necessary to add/remove from the Pool to make Δ zero:

$$\delta(X) = -\frac{\Gamma(t^{\text{arb}})X(t^{\text{arb}}) - Z(t^{\text{arb}})}{2\Gamma(t^{\text{arb}})}$$
$$\delta(Z) = -\delta(X)\Gamma(t^{\text{arb}})$$

Given that before the arbitrage occurs, the pool holds quantities $X(t^{\text{arb}})$ and $Z(t^{\text{arb}})$ of tokens. What interests us is the notional value in USD necessary to perform this rebalancing of the Pool: $N = \Phi(S_{\text{pool}}, \Gamma(t))$. Knowing that all transactions take place at the rate $\Gamma(t)$, we can rearrange $\delta(X)$ and $\delta(Z)$ to make them dependent on S_{pool}, thus obtaining an explicit expression for Φ:

$$\Phi = \left| \frac{\Gamma(t^{\mathrm{arb}})^2[-(X_t + Z_t)] + \Gamma(t^{\mathrm{arb}})[2Z_t - S_{\mathrm{pool}}(Z_t + X_t))]}{2(S_{\mathrm{pool}} + \Gamma(t^{\mathrm{arb}})} \right|$$

with Z_t and X_t being the quantities of tokens present in the Pool before the arbitrage occurs. Considering that the quantities right after the previous `BlockInterval` were $X_t = Z_t = 1$, the modeling of the notional value of intentional arbitrage is then shown in Figure 3.

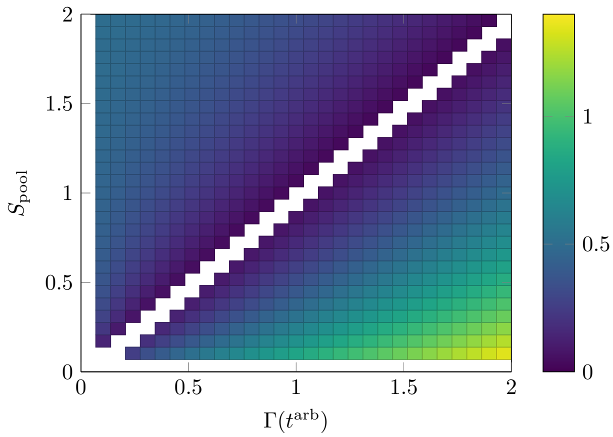

Figure 2: Modélisation de Φ. Lien Modélisation Dynamique.

It is interesting to note that the evolution of ϕ is not linear and is naturally more sensitive to the variation of $\Gamma(t^{\mathrm{arb}})$. We also observe that it is entirely possible that a convergence of scenarios could largely or even completely reduce the notional value required to perform arbitrage.

In this section, we have provided the intuition that the determination of parameters should not be based solely on their impact on the returns of Liquidity Providers. We conclude by proposing a brief observation on the utility of AMMs in general and demonstrate that our model maximizes it.

7 What's the point with AMMs ?

For a market structure, it might be interesting to define a metric μ as the percentage of the total volume generated by market makers that allows other participants, who are not market makers, to trade securities. In our model, the volume of active participants necessary for the proper functioning of the AMM is called IntentionalArbVolume, which is clearly defined as the volume of transactions compensated by `IntentionalArbReward`, denoted as V_{arb}. In defining this metric, we need to consider that the AMM requires the intervention of passive participants, who are exposed to a certain market risk.

$$\mu = \frac{V_{arb}}{V} e^{-\theta t} + E$$

With θ as a parameter capturing the risk incurred by passive participants and their passive nature, and E as a parameter capturing the environment in which the market structure exists. For example, if the environment is Ethereum, then the high block time will be taken into account.

In our model, the market makers (defined by the intentional arbitrageurs) can perfectly predict and estimate the variations in their inventories, a consequence of the pool's parameterization. In the Market-Making model of Grossman and Miller[8], γ is the parameter capturing the penalty for risk-taking by the market maker (risk-aversion parameter). Indeed, in our model, this parameter would be considerably reduced in the sense that it would be easy to bound the movements in the market maker's inventory. Additionally, their PnL (excluding inventory) would be entirely independent of market risk (bid-ask) and compensated by a fixed rate: `IntentionalArbReward`.

For traditional AMM models like Uniswap, a study shows (ref) that their utility, as we have defined it, is less than 1, primarily due to i) The presence of MEV attacks and arbitrageurs (each transaction generates an arbitrage opportunity), which is also extremely toxic for liquidity providers. ii) Liquidity providers in concentrated liquidity AMMs are intended to be active participants, which intrinsically increases their market risk, and therefore reduces θ.

By drastically limiting intentional arbitrage defined as a remunerated transaction, our model aims to maximize μ, and increases the value of θ by reducing the risk for LPs.

References

[1] Conor McMenamin, *An AMM minimizing user-level extractable value and loss-versus-rebalancing.* Department of Information and Communication Technologies, Universitat PompeuFabra, Barcelona, Spain, 2023

[2] Andrea Canidio, *Arbitrageurs' profits, LVR, and sandwich attacks: batch trading as an AMM design response.* ETH Zurich, 2024

[3] Alvaro Cartea, *Decentralised Finance and Automated Market Making: Predictable Loss and Optimal Liquidity Provision.* Oxford-Man Institute of Quantitative Finance, Oxford, UK, 2024

[4] Jason Milionis, *Automated Market Making and Loss-Versus-Rebalancing.* Columbia University, 2023

[5] Masaaki Fukasawa, *Weighted variance swaps hedge against impermanent loss.* Taylor and Francis, 2023

[6] Vijay Mohan, *Automated market makers and decentralized exchanges: a DeFi primer*. Financial Innovation, 2022

[7] Andreas A Aigner, *UNISWAP: Impermanent Loss and Risk Profile of a Liquidity Provider*. Fuel Ventures, 2022

[8] Sanford J. Grossman, *The Informational Role of Upstairs and Downstairs Trading*. he Journal of Business , Vol. 65, No. 4 (Oct., 1992), pp. 509-528, 1992