# am-AMM: An Auction-Managed Automated Market Maker*

Austin Adams[1], Ciamac C. Moallemi[2,3], Sara Reynolds[4], and Dan Robinson[3]

[1] Whetstone Research austin@whetstone.cc
[2] Columbia University ciamac@gsb.columbia.edu
[3] Paradigm dan@paradigm.xyz
[4] Uniswap Labs sara@uniswap.org

**Abstract.** Automated market makers (AMMs) have emerged as the dominant market mechanism for trading on decentralized exchanges implemented on blockchains. This paper presents a single mechanism that targets two important unsolved problems for AMMs: reducing losses to informed orderflow, and maximizing revenue from uninformed orderflow. The "auction-managed AMM" works by running a censorship-resistant onchain auction for the right to temporarily act as "pool manager" for a constant-product AMM. The pool manager sets the swap fee rate on the pool, and also receives the accrued fees from swaps. The pool manager can exclusively capture some arbitrage by trading against the pool in response to small price movements, and also can set swap fees incorporating price sensitivity of retail orderflow and adapting to changing market conditions, with the benefits from both ultimately accruing to liquidity providers. Liquidity providers can enter and exit the pool freely in response to changing rent, though they must pay a small fee on withdrawal. We prove that under certain assumptions, this AMM should have higher liquidity in equilibrium than any standard, fixed-fee AMM.

## 1 Introduction

Liquidity providers (LPs) for automated market makers (AMMs) want to minimize their losses to arbitrageurs while maximizing fee revenue from retail flow. Each of these is a major unsolved problem in AMM design. Minimizing losses to arbitrageurs (which can be characterized as "loss-vs-rebalancing," or LVR) requires either setting high fees or relying on some other mechanism to capture the profits from information or latency arbitrage and mitigate losses to agents (arbitrageurs), who possess superior information on the market value of the asset and snipe stale prices posted by the AMM. Meanwhile, the optimal fee for a given asset pair depends on how retail volume for that pair responds to fees — a difficult problem that is not easy to model. Moreover, the optimal fee may be

* The second author is supported by the Briger Family Digital Finance Lab at Columbia Business School, and is an advisor to Paradigm and to fintech companies. The authors wish to thank Agostino Capponi, Mallesh Pai, and Anthony Zhang for helpful comments.

dynamic and vary with other market variables such as volatility or overall market volume. In standard fixed-fee AMMs (ff-AMMs), these problems compound and interfere with each other: liquidity providers must choose a pool with fees high enough to reduce arbitrage opportunities while still low enough to capture value from retail flow. Additionally, liquidity providers must make this choice statically and for themselves, and if they disagree, liquidity ends up fragmented across multiple pools.

In this paper, we propose the auction-managed AMM (am-AMM), a new AMM design that targets both LVR reduction and fee optimization with one mechanism. The mechanism incentivizes sophisticated market participants to capture some of the value leaked by the AMM, and also lets those market participants set fees at a level that optimizes revenue from retail traders. The pool maintains synchronous composability with other onchain contracts, does not require price oracles, and is resistant to censorship. It also ensures *accessibility*, meaning that liquidity on the pool can always be traded against with some capped fee. Under certain assumptions, we prove that the am-AMM will attract more liquidity than *any* fixed-fee constant product AMM pool (with any fee up to the cap), in equilibrium.

The am-AMM is a constant-product AMM. The AMM utilizes an onchain censorship-resistant "Harberger lease" auction to find the highest bidder. The current highest bidder in the ongoing auction, known as the *manager*, pays rent to liquidity providers. In exchange, the manager can dynamically set the swap fee rate (up to some maximum cap), and receives all swap fees collected by the pool. This allows the manager to capture small arbitrage opportunities each block by trading on the pool, since they can trade with effectively zero fee. It also incentivizes them to set the swap fee in order to maximize revenue from uninformed flow. Liquidity providers can enter and exit the pool freely (though they pay a small withdrawal fee).

The am-AMM has some drawbacks. It may provide even fewer protections against sandwich attacks than a fixed-fee AMM, since the pool manager's ability to trade without fees allows them to profit by pushing any publicly visible transaction to its limit price. It also could contribute to centralization of block builder infrastructure. We think these limitations and drawbacks warrant future study, as does the problem of making this feature work with concentrated liquidity [4].

**Related literature.** While automated market makers are newer compared to many financial market primitives, the literature on the subject is growing rapidly. The concept of AMMs can be traced back to [16] and [26]. Early literature on the current implementations of AMMs includes [7], [6], [20], [10], and [17]. Implementation details of automated market makers are described in [3] and [4]. Furthermore, [2] describes a forthcoming platform for customizable AMMs.

Our paper builds directly on the loss-vs-rebalancing framework established in [23] and [22] as a model for evaluating designs for AMMs. Contemporaneous with the this paper, [21] analyze liquidity provision in AMMs with a similar model of LP profitability and noise trader demand as the present paper.

The idea described in this paper is an instance of an *ex ante* auction for the right to capture the arbitrage profit from the block, an idea first proposed by Alex Herrmann from Gnosis as the "MEV capturing AMM" (McAMM) in [19]. This paper extends that concept to allow the manager to also select the fee charged to retail traders and collect those fees, thus using a similar mechanism to address an independent problem. Also, unlike the McAMM, the auction proposed here guarantees the property of *accessibility* — people can trade against the pool even if the current manager has not submitted a transaction in this block.

Our paper also proposes a way to optimize fees that takes into account the difficult-to-model demand function for retail orderflow. Some AMMs, such as Uniswap v3 [4] address this problem by letting liquidity providers choose between different static "fee tiers." This puts the responsibility for optimizing fees on individual liquidity providers, can lead to some fragmentation of liquidity. Other automated market makers have implemented dynamic fees on individual pools, including Trader Joe v2.1 [24], Curve v2 [13], and Mooniswap [9], as well as [25]. [11] propose a design involving dynamic fees and dynamic price impact functions. While innovative, these dynamic fee implementations are not necessarily optimized for maximizing liquidity. The am-AMM is a first attempt at a provably incentive-compatible dynamic fee, where the fee is set by the market in a way that should always attract more liquidity than any fixed-fee AMM.

## 2 Auction Design

The auction is designed to set up a censorship-resistant version of the two-stage game modelled in Section 3.2, where (1) potential managers bid for the right to set and earn the trading fee in a future block; and (2) liquidity providers respond by adding or removing liquidity. The auction uses a delay parameter $K$. The rules of the auction are designed so that both the rent and the pool manager for block $N$ are locked in as of block $N - K$.

**Harberger lease.** The right to be the manager of a pool is set in a special onchain auction we call a "Harberger lease" (named after the "Harberger taxes" popularized by [27]). This is a continuously held English auction where bids are expressed in terms of rent per block. The *top bid* at any given time determines the manager for the pool, who pays the rent to liquidity providers in the pool for as long as that bid is active.

The rent is denominated and paid in pool tokens, and is paid out of the pool manager's deposit to all pool LP token holders proportional to their stake (thus effectively increasing the value of those pool tokens). This ensures that pool tokens remain fungible and that rent is compounded automatically.

**Bidding rules.** Each bid specifies a per-block rent $R$. When the bid is placed, it must include a *deposit* $D$, which must be a multiple of $R$ and must be at least $R \cdot K$. Newly placed bids do not become active immediately, but are delayed by $K$ blocks. When either a new high bid becomes active, or the current pool manager's deposit is depleted, the new high bidder *usurps* the current pool manager. The contract enforces a minimum bid increment.

The top bid cannot be cancelled, but can reduce its deposit as long as it leave a minimum deposit of $D_{\text{top}} \geq R_{\text{top}}K$. The contract keeps track of the *next* bid (which is either the next-highest bid after the current top bid, or a bid that is higher than the top bid but is not yet active). If the top bid does not have enough rent to pay for $K$ blocks — that is, if $\frac{D_{\text{top}}}{R_{\text{top}}} < K$, then the next bid cannot be canceled, but must leave at least enough deposit such that $\frac{D_{\text{top}}}{R_{\text{top}}} + \frac{D_{\text{next}}}{R_{\text{next}}} \geq K$.

**Pool manager rights.** The current pool manager can adjust the swap fee for the next block at any time, subject to a fixed *fee cap* $f_{\max}$. The pool manager also receives all swap fees collected by the pool. Since the pool manager receives swap fees, they are effectively able to swap on the pool with zero fee. This means that they can capture arbitrages from small price movements that no other arbitrageur would be able to profitably capture. However, if the price moves by more than the current fee in a single block — in other words, if it moves outside the "no-trade region" [22] — then some of the profit could leak to other arbitrageurs, as discussed in Section 3.2.

**Censorship resistance.** The delay parameter $K$ should be chosen such that there is a negligible probability that anyone can censor the base layer for $K$ blocks in a row. This prevents someone from stealing MEV from a high-volatility block by usurping the current manager and then censoring arbitrage transactions for $K$ blocks. It also ensures that liquidity providers can respond to changes in rent by adding or removing liquidity before the new rent takes effect.

**Withdrawal fees.** Liquidity providers are free to enter or exit at any time. This allows them to respond to anticipated changes in rent during the $K$-block delay. However, when liquidity providers exit, they must pay a small withdrawal fee to the current manager. This prevents liquidity providers from withdrawing liquidity after volatility is realized but before the current manager has the opportunity to execute an arbitrage transaction. As shown in Appendix B, even a very small withdrawal fee — less than 0.13 basis points, if the fee cap is 1% — can ensure that strategic liquidity provider withdrawals cannot reduce the manager's profit from a given arbitrage opportunity below the amount assumed in Section 3. This fee could alternatively be replaced with a withdrawal delay for a similar effect.

## 3 Theory

Our starting point is a model inspired by [23] and [22], which consider an AMM trading a risky asset (denoted by $x$) versus the numéraire (denoted by $y$), and that the risky asset has a fundamental price at all times (for example, on an infinitely deep centralized exchange). As in [22], we assume that traders can only trade on the pool at discrete block generation times.

Our setting is consistent with and can be structurally microfounded in the full setting of [22], which assumes that blocks arrival times follow a Poisson process and that the asset's price follows geometric Brownian motion parameterized by volatility $\sigma > 0$, but our theorems do not depend on those assumptions. Instead

we describe the model primitives in reduced form, with weaker assumptions, and provide examples that show that those models would satisfy the assumptions.

We will restrict to the case of a constant product market maker,[5] with invariant $\sqrt{xy} = L$, where we denote the reserve quantities by $(x, y)$ and the pool liquidity level by $L$. If the price of the risky asset is given by $P$, the value of the the pool reserves is given by $V(L) \triangleq 2\sqrt{P}L$, as a function of the available liquidity $L$. We assume the pool charges a proportional trading fee $f \in [0, f_{\max}]$.

We consider a setting where there are two types of traders: (1) noise traders, who trade for idiosyncratic reasons and generate fee income for the pool; and (2) arbitrageurs, who seek to exploit price differences between the pool and the fundamental price, and create adverse selection costs for the pool. Here, the P&L of LPs is a jump process, stochastically jumping at instances of block generation. We will consider the expected aggregate instantaneous rate of P&L of LPs per unit time, where we are averaging over stochasticity in future price changes, or, equivalently, assuming that exposure to market risk of the risky asset has been hedged.

**Noise traders.** We assume there exists a population of noise traders that trade for idiosyncratic reasons (e.g., convenience of executing on chain) and not for informational reasons. Hence, economically, in our model noise traders serve exclusively to generate fee income for the pool. Given pool fee $f \geq 0$ and liquidity $L \geq 0$, denote by $H(f, L)$ the expected total volume of noise trades arriving per unit time, denominated in the numéraire, so that $fH(f, L)$ is the total rate per unit time of fee revenue generated by the noise traders. We will assume that:

**Assumption 1 (Noise trader demand)** *For $L > 0$, define*

$$H_0(f, L) \triangleq H(f, L)/V(L),$$

*to be the expected noise trader volume per unit time per unit of pool value, we assume that $H_0(\cdot, \cdot)$ is a continuous function satisfying:*

1. *For all $L > 0$, $H_0(f, L)$ is a decreasing function of $f$.*
2. *For all $f \in [0, f_{\max}]$, $H_0(f, L)$ is strictly decreasing function of $L$. Moreover, $H_0(f, L) \downarrow 0$ as $L \to \infty$, and $H_0(f, L) \uparrow \infty$ as $L \to 0$.*

Part 1 asserts that noise trader demand is decreasing in the price (fee) charged. Part 2 implies that the noise trader is sub-linear in pool value or liquidity, i.e., noise trader demand is satiated.

[5] The key property of the constant product market maker we use is that the arbitrage profits and arbitrage excess scale linearly with pool value, where the constant of proportionality does not depend on the price. This property holds more generally for geometric mean market makers, and our results would trivially extend there. Beyond that, when considering more general invariant curves, there may be additional second order effects due to the price changing over time horizon of the LP investment. If price movements over the scale of the LP investment horizon are not large, these effects may not be significant, and we would expect the high level insights of our model to continue to apply to general invariant curves.

*Example 1.* Consider $H(f, L) \triangleq c_0 L^\alpha \exp(-c_1 f)$, where $c_0, c_1 > 0$ and $\alpha \in (0, 1)$ are parameters that could be estimated from transaction data, and may be time varying or may depend on other market parameters such as volatility or broader market volume. This liquidity dependence is consistent with the model proposed by [18].

**Arbitrageur profits.** We assume there is a competitive and deep market of arbitrageurs monitoring prices in the AMM and exploiting price differences between the AMM and the fundamental price. Denote by $\mathsf{ARB\_PROFIT}(f, L)$ the expected profit to arbitrageurs per unit time, given fee $f \in [0, f_{\max}]$ and liquidity $L \geq 0$. We make the following assumption:

**Assumption 2 (Arbitrageur profits)** *For $L > 0$, we assume that*

$$\mathsf{ARB\_PROFIT}(f, L) = \mathsf{AP}_0(f) V(L),$$

*where $\mathsf{AP}_0(\cdot)$ is a continuous, decreasing function.*

Assumption 2 guarantees that arbitrage profits scale *linearly* with the pool value or liquidity, in contrast to noise trader volume. This is because we assume, by nature of engaging in riskless activity, arbs have access to infinite capital for arbitrage activity, and arbitrage demand will not be satiated so long as arbitrage opportunities are available. Assumption 2 also implies that, holding liquidity fixed, arbitrageur profits are decreasing in the fee $f$, this is because the fee is a friction that limits arbitrage.

This assumption does not depend on a particular model for either the asset price's behavior or for block generation times, but we show that it can be structurally microfounded in one such model, as illustrated here:[6]

*Example 2.* Consider a setting where the risky asset's price follows geometric Brownian motion parameterized by volatility $\sigma > 0$ and blocks are generated at the arrivals of a Poisson process of rate $\Delta t^{-1}$, with $\Delta t$ being the average interblock time. [22] establish that, when $\Delta t < 8\sigma^2$,

$$\mathsf{ARB\_PROFIT}(f, L) = \frac{\sigma^2}{8} \frac{1}{1 + \frac{f}{\sigma\sqrt{\Delta t/2}}} \frac{e^{+f/2} + e^{-f/2}}{2\Big(1 - \sigma^2 \Delta t/8\Big)} V(L). \tag{1}$$

This satisfies Assumption 2.

### 3.1 Fixed-Fee AMM Model

As a benchmark, we consider the standard AMM design with a fixed fee $f \in [0, f_{\max}]$, which we refer to as a *fixed-fee AMM* (ff-AMM). Following the discussion above, we define the instantaneous rate of aggregate expected P&L of LPs in this pool in excess of the risk free rate according to

$$\begin{aligned} \Pi^{\mathsf{LP}}_{\mathsf{ff}}(f, L) &\triangleq f H(f, L) - \mathsf{ARB\_PROFIT}(f, L) - r V(L) \\ &= \left(f H_0(f, L) - \mathsf{AP}_0(f) - r\right) V(L), \end{aligned} \tag{2}$$

[6] See Section 4 for further discussion.

where the first term in the sum is the revenue from noise traders, the second term is the loss to arbs, and the final term is a capital charge, where $r > 0$ is the risk-free rate. Under free entry and exit of LPs, we define the equilibrium in the fixed-fee AMM according to a zero profit condition:

**Lemma 1 (ff-AMM equilibrium).** *Given fee $f \in [0, f_{\max}]$, define a liquidity level $L^* > 0$ to be a competitive equilibrium if LPs earn zero profit in excess of the risk free rate, i.e., $\Pi^{\mathsf{LP}}_{\mathsf{ff}}(f, L^*) = 0$. Then, for any $f \in [0, f_{\max}]$, a unique equilibrium level of liquidity $L^* = L_{\mathsf{ff}}(f)$ exists.*

### 3.2 Auction-Managed AMM Model

In this section, we consider the auction-managed AMM design.

**Arbitrageur excess.** In the am-AMM, the pool manager collects all of the fee revenue, and therefore, effectively, can also trade against the pool without paying any fees. Therefore, the pool will suffer adverse selection costs of $\mathsf{ARB\_PROFIT}(0, L)$, independent of the fee level $f$. However, the manager does not collect all of these fees as income. Instead, note that whenever the mispricing in the pool exceeds the fee $f$, some of that mispricing can be profitably captured by agents other than the manager. We denote by $\mathsf{ARB\_EXCESS}(f, L)$ the instantaneous rate of expected arbitrage profit per unit time forgone by the manager. We assume that:

**Assumption 3 (Arbitrageur excess)** *For $L > 0$, we assume that*

$$\mathsf{ARB\_EXCESS}(f, L) = \mathsf{AE}_0(f)V(L),$$

*with $\mathsf{AE}_0(f)$ a continuous, decreasing function that satisfies $\mathsf{AE}_0(f) \leq \mathsf{AP}_0(f)$, for all $f \in [0, f_{\max}]$, and where the inequality is strict if $f > 0$.*

Assumption 3 is largely analogous to Assumption 2, and can be similarly microfounded by a stochastic model, as shown in Example 4, but does not depend on that model. The strict inequality assumption simply asserts that *some* of the arbitrage profit is captured by the manager.

*Example 3.* One extreme setting is the McAMM design of [19]. There, no trades can occur in an AMM in a given block unless the pool manager has a transaction earlier in the block to "unlock" the pool. Hence the pool manager is guaranteed to be the first transaction in everyblock and can capture all arbitrage profits. In this case, $\mathsf{ARB\_EXCESS}(f, L) = 0$.

*Example 4.* In Section 4, we develop structural microfoundations of a model for arbitrageur excess in the setting of [22], where the risky asset's price follows geometric Brownian motion parameterized by volatility $\sigma > 0$ and blocks are generated at the arrivals of a Poisson process of rate $\Delta t^{-1}$, with $\Delta t$ being the average interblock time. In that setting, we assume that (1) the pool manager fully monetizes any arbitrage where the mispricing is less than the fee $f$, and (2) when the mispricing exceeds the fee $f$, other arbitrageurs correct the mispricing back to the fee level $f$ before the pool manager can trade, and hence the pool

manager is only able to monetize the portion of the arbitrage up to $f$. Then, we establish that, when $\Delta t < 8\sigma^2$,

$$\mathsf{ARB\_EXCESS}(f, L) = \frac{\sigma^2}{8} \exp\left(-\frac{f}{\sigma\sqrt{\Delta t/2}}\right) \frac{e^{+f/2} + e^{-f/2}}{2\Big(1 - \sigma^2 \Delta t/8\Big)} V(L). \tag{3}$$

This satisfies the conditions of Assumption 3.

Comparing Example 2 and Example 4, we see that $\mathsf{ARB\_EXCESS}(f, L) \ll \mathsf{ARB\_PROFIT}(f, L)$ in the sense that

$$\frac{\mathsf{ARB\_EXCESS}(f, L)}{\mathsf{ARB\_PROFIT}(f, L)} = \frac{\mathsf{AE}_0(f, L)}{\mathsf{AP}_0(f, L)} = \left(1 + \frac{f}{\sigma\sqrt{\Delta t/2}}\right) \exp\left(-\frac{f}{\sigma\sqrt{\Delta t/2}}\right),$$

which is exponentially vanishing in $f/\sigma\sqrt{\Delta t}$.

**Equilibrium.** We imagine a game that proceeds in two steps: (1) agents bid rent $R$, the agent with the highest rent wins the auction and is declared the pool manager, with the right to determine the trading fee $f$ and earn all fees; and (2) LPs determine aggregate liquidity $L$ given $R$. Based on the discussion above, the P&L of the manager per unit time is given by

$$\begin{aligned}\Pi^{\mathsf{MGR}}_{\mathsf{am}}(R, L) &\triangleq \max_{f \in [0, f_{\max}]} \{ f H(f, L) + \mathsf{ARB\_PROFIT}(0, L) \\ &\qquad\qquad - \mathsf{ARB\_EXCESS}(f, L) - R \} \\ &= \max_{f \in [0, f_{\max}]} \{ f H_0(f, L) + \mathsf{AP}_0(0) - \mathsf{AE}_0(f) \} V(L) - R.\end{aligned}$$

The maximization is because the manager is free to set the fee, and we assume they will do so to maximize P&L. The first term captures the fact that the manager retains all fee revenue. The second and third terms capture the fact that the manager earns arbitrage profits as if it pays no fees, except for the arbitrage excess. The final term captures the fact that the manager pays rent. On the other hand, the LPs in aggregate earn P&L per unit time given by

$$\Pi^{\mathsf{LP}}_{\mathsf{am}}(R, L) \triangleq R - \mathsf{ARB\_PROFIT}(0, L) - rV(L) = R - (\mathsf{AP}_0(0) + r)\, V(L).$$

Here, the first term is the rent payment made by the manager, the second term is the adverse selection cost (which is as if no fees are charged), and the third term is the cost of capital. Given these definitions, we have that:

**Theorem 1 (am-AMM equilibrium).** *$(R^*, L^*)$ is a competitive equilibrium of the am-AMM if the zero profit conditions*[7]

$$\Pi^{\mathsf{MGR}}_{\mathsf{am}}(R^*, L^*) = 0, \qquad \Pi^{\mathsf{LP}}_{\mathsf{am}}(R^*, L^*) = 0,$$

[7] Here, we assume that entry and exit are frictionless for LPs, ignoring the withdrawal fees. This is justified if the liquidity provision decision horizon is sufficiently long so that, amortized over its length, the withdrawal fees are de minimus.

*are satisfied. An equilibrium* $(R^*, L^*)$ *must exist, with equilibrium fees*

$$f^* \in \underset{f \in [0, f_{\max}]}{\operatorname{argmax}} \; f H_0(f, L) - \mathsf{AE}_0(f), \tag{4}$$

*and satisfies* $L^* > L_{\mathsf{ff}}(f)$*, for all* $f \geq 0$*. Therefore, in equilibrium, the am-AMM will have higher liquidity than any ff-AMM.*

Theorem 1 establishes that the equilibrium liquidity for the am-AMM is larger than that of any ff-AMM. Beyond that, it gives insight into the equilibrium fee $f^*$ set in the am-AMM: from (4), the pool operator will seek to set the fee to maximize noise trader revenue adjusted for arbitrageur excess. As a comparison, consider the fee level $f_{\mathsf{opt}}$that exclusively maximizes noise trader revenue, i.e.,

$$f_{\mathsf{opt}} \in \underset{f \in [0, f_{\max}]}{\operatorname{argmax}} \; f H_0(f, L^*). \tag{5}$$

If we assume that the revenue function $f \mapsto f H_0(f, L^*)$ is concave, and the arbitrageur excess function $\mathsf{AE}_0(f)$ is convex, then a comparison of first order conditions for (4)–(5) reveals that $f_{\mathsf{opt}} \leq f^*$, i.e., the am-AMM will set fees higher than is purely revenue optimal. However, by virtue of the optimality of $f^*$ in (4), we have that

$$f_{\mathsf{opt}} H_0(f_{\mathsf{opt}}, L^*) - f^* H_0(f^*, L^*) \leq \mathsf{AE}_0(f_{\mathsf{opt}}) - \mathsf{AE}_0(f^*) \leq \mathsf{AE}_0(f_{\mathsf{opt}}). \tag{6}$$

In the arbitrageur excess model of Example 4, $\mathsf{AE}_0(f_{\mathsf{opt}})$ may be very small (since it is exponentially vanishing in blocktime, for example), and in such cases (6) would imply that $f^*$ is also nearly optimal from a noise trader revenue perspective.

## 4 A Structural Model for Arbitrageur Excess

In this section, we will derive the structural model for arbitrageur excess of Example 4, following the arbitrageur profits model of [22] and of Example 2.

**Fee structure.** In order to simplify formulas, [22] uses a fee structure wherein a proportional fee of $e^{+\gamma} - 1 = \gamma + o(\gamma)$ is charged for purchases of the risky asset from the pool, while a proportional fee of $1 - e^{-\gamma} = \gamma + o(\gamma)$ is charged for sales of the risky asset to the pool — these are symmetric fees in log-price space. This is different than the setting in this paper, where we assume a fee proportional $f$ which is the same for buys or sells — that is, symmetric in (linear) price space. In order to facilitate comparison with [22], we will make the approximation $f = \gamma$. As illustrated in Example 1 of [22], for practical parameter values, this does not make a significant difference.

**Asset price dynamics.** Denote by $P_t$ the fundamental price of the asset, which follows a geometric Brownian motion with drift $\mu > 0$ and volatity $\sigma > 0$, and denote by $\tilde{P}_t$ the implied spot price of the asset in the pool. Define $z_t \triangleq \log P_t / \tilde{P}_t$

to be the log-mispricing at time $t$. When $t$ is not a block generation time, Itô's lemma implies that $z_t$ is governed by the stochastic differential equation

$$dz_t = \left(\mu - \tfrac{1}{2}\sigma^2\right)\, dt + \sigma\, dB_t, \tag{7}$$

where $B_t$ is a Brownian motion. We will make the symmetry assumption that $\mu = \frac{1}{2}\sigma^2$, so that $z_t$ is a driftless, (scaled) Brownian motion.

**Block time dynamics.** We assume that blocks are generated according to a Poisson process with mean interarrival time $\Delta t$. Denote the block generation times by $0 < \tau_1 < \tau_2 < \ldots$. At instances $t = \tau_i$ when blocks are generated, we imagine that

1. If the $|z_{t-}| \geq f$ (i.e., the mispricing immediately before block generation exceeds the fee), we imagine an arbitrageur is able to trade until the mispricing is equal to the fee, and thus earns profits that are not captured by the pool operator. These profits then become part of arbitrageur excess. Following the derivation of [22], the instaneous profit from arbitrageur excess when price is $P = P_t$ and the mispricing is $z = z_{t-}$ due to buying (respectively, selling) is given by

$$A_+(P,z) \triangleq \left[P\left\{x^*\left(Pe^{-z}\right) - x^*\left(Pe^{-f}\right)\right\} + e^{+f}\left\{y^*\left(Pe^{-z}\right) - y^*\left(Pe^{-f}\right)\right\}\right]\mathbb{I}_{\{z>+f\}} \geq 0,$$

$$A_-(P,z) \triangleq \left[P\left\{x^*\left(Pe^{-z}\right) - x^*\left(Pe^{+f}\right)\right\} + e^{-f}\left\{y^*\left(Pe^{-z}\right) - y^*\left(Pe^{+f}\right)\right\}\right]\mathbb{I}_{\{z<-f\}} \geq 0.$$

   Here, the holdings of the pool when implied price is $P$ are given by $x^*(P) \triangleq L/\sqrt{P}$, $y^*(P) = L\sqrt{P}$.
2. After the arbitrageur trades, the pool operator corrects any remaining mispricing, so that $z_t = 0$.

**Intensity of arbitrageur excess.** The intensity or instaneous rate of arbitrageur excess per dollar of pool value per unit time is given by

$$\mathsf{AE}_0(f) = \frac{\mathsf{ARB\_EXCESS}(f,L)}{V(L)} = \frac{1}{\Delta t}\mathsf{E}\left[\frac{A_+(P,z) + A_-(P,z)}{V(L)}\right],$$

where the expectation is over $z = z_\tau$ at the next block generation time $\tau$, i.e., $z \sim N(0, \sigma^2\tau)$ with $\tau \sim \mathrm{Exp}(\Delta t^{-1})$.

Observe that

$$\begin{aligned}
\frac{A_+(P,z)}{V(L)} &= \frac{1}{2L\sqrt{P}}\left[P\left\{x^*\left(Pe^{-z}\right) - x^*\left(Pe^{-f}\right)\right\} + e^{+f}\left\{y^*\left(Pe^{-z}\right) - y^*\left(Pe^{-f}\right)\right\}\right]\mathbb{I}_{\{z>+f\}} \\
&= \tfrac{1}{2}\left[\left\{e^{+z/2} - e^{+f/2}\right\} + e^{+f}\left\{e^{-z/2} - e^{-f/2}\right\}\right]\mathbb{I}_{\{z>+f\}} \\
&= \tfrac{1}{2}e^{+f/2}\left[e^{+(z-f)/2} - 2 + e^{-(z-f)/2}\right]\mathbb{I}_{\{z>+f\}}, \\
\frac{A_-(P,z)}{V(L)} &= \frac{1}{2L\sqrt{P}}\left[P\left\{x^*\left(Pe^{-z}\right) - x^*\left(Pe^{+f}\right)\right\} + e^{-f}\left\{y^*\left(Pe^{-z}\right) - y^*\left(Pe^{+f}\right)\right\}\right]\mathbb{I}_{\{z<-f\}} \\
&= \tfrac{1}{2}\left[\left\{e^{+z/2} - e^{-f/2}\right\} + e^{-f}\left\{e^{-z/2} - e^{+f/2}\right\}\right]\mathbb{I}_{\{z<-f\}} \\
&= \tfrac{1}{2}e^{-f/2}\left[e^{+(z+f)/2} - 2 + e^{-(z+f)/2}\right]\mathbb{I}_{\{z<-f\}},
\end{aligned}$$

Taking expectations of the first term, conditioned on the block generation time $\tau$, we have $z \sim N(0, \sigma^2\tau)$, so that

$$\mathsf{E}\left[\left.\frac{A_+(P,z)}{V(L)}\right|\ \tau\right] = -e^{\frac{f}{2}} + \frac{1}{2}\left(e^f+1\right)e^{\frac{\sigma^2\tau}{8}} - \frac{1}{2}e^f e^{\frac{\sigma^2\tau}{8}}\operatorname{erf}\left(\frac{\sigma^2\tau+2f}{2\sqrt{2}\sigma\sqrt{\tau}}\right) + \frac{1}{2}e^{\frac{\sigma^2\tau}{8}}\operatorname{erf}\left(\frac{\sigma^2\tau-2f}{2\sqrt{2}\sigma\sqrt{\tau}}\right) + e^{\frac{f}{2}}\operatorname{erf}\left(\frac{f}{\sigma\sqrt{\tau}\sqrt{2}}\right).$$

Taking expectations over $\tau \sim \mathrm{Exp}(\Delta t^{-1})$, assuming that $\Delta t < 8\sigma^2$,

$$\frac{1}{\Delta t}\mathsf{E}\left[\frac{A_+(P,z)}{V(L)}\right] = \frac{\sigma^2}{8}\exp\left(-\frac{f}{\sigma\sqrt{\Delta t/2}}\right)\frac{e^{+f/2}}{2\Big(1-\sigma^2\Delta t/8\Big)}.$$

Similarly, for the other term,

$$\frac{1}{\Delta t}\mathsf{E}\left[\frac{A_-(P,z)}{V(L)}\right] = \frac{\sigma^2}{8}\exp\left(-\frac{f}{\sigma\sqrt{\Delta t/2}}\right)\frac{e^{-f/2}}{2\Big(1-\sigma^2\Delta t/8\Big)}.$$

Combining these results yields the arbitrageur excess expression in (3).

**Discussion.** The expressions (1) for arbitrageur profits and (3) for arb excess have an interesting common structure. In both cases, the expressions can be decomposed into the product of (1) the probability that an arbitrageur can trade at the next block time; and (2) the expected profit conditioned on trade. In the arbitrageur profits expression, the probability of trade is given by

$$\frac{1}{1+\frac{f}{\sigma\sqrt{\Delta t/2}}},$$

while in the arbitrageur excess expression, it is given by

$$\exp\left(-\frac{f}{\sigma\sqrt{\Delta t/2}}\right).$$

In both cases, however, the expected profit conditioned on trade is the same. Thus, the fact that arbitrageur excess is less than arbitrageur profits is driven by the fact that the probability of trade is (exponentially) less. This, in turn, is because the pool operator drives the mispricing to zero at the end of every block, making large mispricings at the beginning of the next block unlikely.

## 5 Discussion

**Advantages.** One desirable characteristic of the auction-managed AMM is that it shifts the strategic burden of determining the optimal fee from passive LPs to the pool manager, who, in turn will set the fee according to (4), which seeks

to set fees to optimize revenue from noise traders, adjusted by arb excess paid to arbitraguers. This makes sense, since the pool manager is assumed to be a more sophisticated entity that can perform off-chain modeling and analysis to estimate noise trader sensitivity and losses to arbs.

Our model captures the most visible and salient features that drive noise trader demand (the fee and the liquidity). However, as argued by [22] and [28], noise traders may also be concerned with the accuracy of prices in the pool. Relative to an outside reference price (e.g., the price on infinitely deep centralized exchange), noise traders pay an effective spread which is the sum of the trading fee of the pool (deterministic, positive) and the relative mispricing of the pool (stochastic, could be positive or negative). In general noise traders will prefer pools with less relative mispricing. An additional benefit of the am-AMM over the ff-AMM is that the quoted prices are more accurate. This is because the pool operator of an am-AMM faces no fees in performing arbitrage trades against the pool, and is able to correct smaller price discrepancies.

There is also an important transfer of risk between the LPs and the pool manager: in the am-AMM, the LPs earn rent payments instead of noise trader fees. Since these rent payments are determined *ex ante*, in equilibrium, they incorporate the expected value of future noise trader fee revenue, in contrast to the actual noise trader fee revenue, which arrives in a lumpy and stochastic fashion and goes to the pool manager. Although it is beyond the scope of the analysis we have presented (which assumes risk neutrality), this risk transfer is likely welfare improving since the pool manager is likely a larger and better capitalized entity than a typical passive LP, and is thus less risk averse.

**Drawbacks.** The auction-managed AMM is not without drawbacks. First, the pool manager's ability to effectively trade on the pool with zero spread exacerbates the "sandwich attack" problem with onchain AMMs, as discussed generally by [12,29,1]. A party that can trade with zero fees can profit by pushing any publicly visible swap transaction to its limit price. This is a meaningful drawback — which could hurt swappers or ultimately discourage retail flow, thus potentially invalidating the assumption in Section 3 that retail flow is only a function of liquidity and fee — but known mitigations for general sandwich attacks could be applied, such as verifiable sequencing rules [14], private relays [1], or offchain filler auctions [5].

Second, the *ex ante* auction for the arbitrage opportunity could have effects on the market for block building. The pool manager's exclusive right to capture some arbitrage profit from a pool could give them an advantage in the block builder auction, similar to the advantages enjoyed by builders with private orderflow discussed by [15]. Additionally, while the am-AMM allows any party to bid on the pool manager position, parties with greater sophistication or access to private orderflow may be more likely to win the auction. On the other hand, by capturing some arbitrage profit and reducing the power of block proposers, the am-AMM could mitigate some of the harmful pressures that MEV puts on the blockchain consensus mechanism. We think these possible consequences warrant further study.

Lastly, one disadvantage of the am-AMM over the McAMM discussed by [19] is that the current manager does not capture the "arbitrage excess" that results from single-block price movements beyond the no-trade region, as discussed in Section 3. This creates an incentive for the manager to set a higher fee than the one that optimizes total fee revenue. This could be fixed by requiring the manager to unlock the pool at every block, at the cost of sacrificing the accessibility property.

**Future work.** In this paper, we focus on constant product market makers. Extending this design to other AMMs — and particularly to concentrated liquidity AMMs, in which different liquidity providers may have different returns during the same period and may go in or out of range as a result of price changes — is left for future work.

Implementing this mechanism is also left for future work. Since we first made this paper available, there has been at least one open-source third-party implementation [8].

## References


1. Austin Adams, Benjamin Y Chan, Sarit Markovich, and Xin Wan. The costs of swapping on the uniswap protocol. *arXiv preprint arXiv:2309.13648*, 2023.
2. Hayden Adams, Moody Salem, Noah Zinsmeister, Sara Reynolds, Austin Adams, Will Pote, Mark Toda, Alice Henshaw, Emily Williams, and Dan Robinson. Uniswap v4 core [draft], 2023. URL: `https://github.com/Uniswap/v4-core/blob/main/docs/whitepaper-v4.pdf`.
3. Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core, March 2020. URL: `https://uniswap.org/whitepaper.pdf`.
4. Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core, March 2021. URL: `https://uniswap.org/whitepaper-v3.pdf`.
5. Hayden Adams, Noah Zinsmeister, Mark Toda, Emily Williams, Xin Wan, Matteo Leibowitz, Will Pote, Allen Lin, Eric Zhong, Zhiyuan Yang, Riley Campbell, Alex Karys, and Dan Robinson. Uniswapx, 2023. URL: `https://uniswap.org/whitepaper-uniswapx.pdf`.
6. Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 80–91, 2020.
7. Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. *arXiv preprint arXiv:1911.03380*, 2019.
8. BidDog. Biddog: Open source implementation of am-amm auctions, 2024. Accessed: 2024-05-22. URL: `https://github.com/Bunniapp/biddog`.
9. Anton Bukov and Mikhail Melnik. Mooniswap by 1inch.exchange, August 2020. URL: `https://mooniswap.exchange/docs/MooniswapWhitePaper-v1.0.pdf`.
10. Agostino Capponi and Ruizhe Jia. The adoption of blockchain-based decentralized exchanges. *arXiv preprint arXiv:2103.08842*, 2021.
11. Álvaro Cartea, Fayçal Drissi, Leandro Sánchez-Betancourt, David Siska, and Lukasz Szpruch. Automated market makers designs beyond constant functions. *Available at SSRN 4459177*, 2023.

12. Philip Daian, Steven Goldfeder, Tyler Kell, Yunqi Li, Xueyuan Zhao, Iddo Bentov, Lorenz Breidenbach, and Ari Juels. Flash boys 2.0: Frontrunning in decentralized exchanges, miner extractable value, and consensus instability. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 910–927. IEEE, 2020.
13. Michael Egorov and Curve Finance (Swiss Stake GmbH). Automatic market-making with dynamic peg, June 2021. URL: `https://classic.curve.fi/files/crypto-pools-paper.pdf`.
14. Matheus VX Ferreira and David C Parkes. Credible decentralized exchange design via verifiable sequencing rules. *arXiv preprint arXiv:2209.15569*, 2022.
15. Tivas Gupta, Mallesh M Pai, and Max Resnick. The centralizing effects of private order flow on proposer-builder separation. *arXiv preprint arXiv:2305.19150*, 2023.
16. Robin Hanson. Logarithmic markets coring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15, 2007.
17. Joel Hasbrouck, Thomas J Rivera, and Fahad Saleh. The need for fees at a dex: How increases in fees can increase dex trading volume. *Available at SSRN*, 2022.
18. Joel Hasbrouck, Thomas J. Rivera, and Fahad Saleh. An economic model of a decentralized exchange with concentrated liquidity. Working paper, 2023.
19. Alex Herrmann. Mev capturing amm (mcamm), August 2022. URL: `https://ethresear.ch/t/mev-capturing-amm-mcamm/13336`.
20. Alfred Lehar and Christine A Parlour. Decentralized exchanges. *Available at SSRN 3905316*, 2021.
21. Julian Ma and Davide Crapis. The cost of permissionless liquidity provision in automated market makers. *arXiv preprint arXiv:2402.18256*, 2024.
22. Jason Milionis, Ciamac C Moallemi, and Tim Roughgarden. Automated market making and arbitrage profits in the presence of fees. *arXiv preprint arXiv:2305.14604*, 2023.
23. Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing, 2022. URL: `https://arxiv.org/abs/2208.06046`, `doi:10.48550/ARXIV.2208.06046`.
24. MountainFarmer, Louis, Hanzo, Wawa, Murloc, and Fish. Joe v2.1 liquidity book, November 2022. URL: `https://github.com/traderjoe-xyz/LB-Whitepaper/blob/main/Joe%20v2%20Liquidity%20Book%20Whitepaper.pdf`.
25. Alex Nezlobin. Twitter thread, June 2023. URL: `https://twitter.com/0x94305/status/1674857993740111872`.
26. Abraham Othman, David M Pennock, Daniel M Reeves, and Tuomas Sandholm. A practical liquidity-sensitive automated market maker. *ACM Transactions on Economics and Computation (TEAC)*, 1(3):1–25, 2013.
27. Eric A. Posner and E. Glen Weyl. *Radical Markets: Uprooting Capitalism and Democracy for a Just Society*. Princeton University Press, Princeton, NJ, 2018.
28. Rithvik Rao and Nihar Shah. Triangle fees, 2023. `arXiv:2306.17316`.
29. Liyi Zhou, Kaihua Qin, Christof Ferreira Torres, Duc V Le, and Arthur Gervais. High-frequency trading on decentralized on-chain exchanges. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 428–445. IEEE, 2021.

# A Proofs

**Proof** *(Proof of Lemma 1).* Given fixed $f \in [0, f_{\max}]$, define $G(L) \triangleq fH_0(f, L) - \mathsf{AP}_0(f) - r$. By Assumption 1(2) and Assumption 2, this is a continuous function, and

$$\lim_{L \to 0} G(L) = \infty, \quad \lim_{L \to \infty} G(L) = -\mathsf{AP}_0(f) - r < 0.$$

By the intermediate value theorem, there exists $L^* > 0$ with $G(L^*) = 0$, comparing with (2), this must be an equilibrium. This equilibrium is unique since $G(\cdot)$ is strictly monotonic.

**Proof** *(Proof of Theorem 1).* First, we will prove that the equilibrium $(R^*, L^*)$ exists. Solving for $R^*$ in the condition $\Pi^{\mathsf{MGR}}_{\mathsf{am}}(R^*, L^*) = 0$ and substituting into $\Pi^{\mathsf{LP}}_{\mathsf{am}}(R^*, L^*) = 0$, we have that $L^*$ must satisfy $G(L^*) = 0$ where

$$G(L) \triangleq \max_{f \in [0, f_{\max}]} \; fH_0(f, L) - \mathsf{AE}_0(f) - r.$$

From Assumption 1 and Assumption 2, this function is continuous, and

$$\lim_{L \to \infty} G(L) = \max_{f \in [0, f_{\max}]} \; -\mathsf{AE}_0(f) - r = -\mathsf{AE}_0(0) - r < 0.$$

$$\lim_{L \to 0} G(L) \geq \lim_{L \to 0} fH_0(f, L) - \mathsf{AE}_0(f) - r = \infty > 0,$$

for any $f \in (0, f_{\max}]$. By the intermediate value theorem, an equilibrium $(R^*, L^*)$ must exist.

In order to compare with the ff-AMM, define

$$L_{\max} \triangleq \max_{f \in [0, f_{\max}]} \; L_{\mathsf{ff}}(f),$$

to be the maximum level of liquidity that can be achieved in the ff-AMM, and denote by $f^*$ the maximizing fee. Define

$$R_{\max} \triangleq (\mathsf{AP}_0(0) + r) V(L_{\max}).$$

From the zero profit condition $\Pi^{\mathsf{LP}}_{\mathsf{am}}(R_{\max}, L_{\max}) = 0$, it is clear that a rent payment of $R_{\max}$ will incentivize liquidity $L_{\max}$. Under such a rent payment, the pool manager earns profits

$$\begin{aligned}
\Pi^{\mathsf{MGR}}_{\mathsf{am}}(R_{\max}, L_{\max}) &= \left( \mathsf{AP}_0(0) + \max_{f \in [0, f_{\max}]} \; fH_0(f, L_{\max}) - \mathsf{AE}_0(f) \right) V(L_{\max}) - R_{\max} \\
&\geq \left( \mathsf{AP}_0(0) + f^* H_0(f^*, L_{\max}) - \mathsf{AE}_0(f^*) \right) V(L_{\max}) - R_{\max} \\
&= \left( \mathsf{AP}_0(0) + \mathsf{AP}_0(f^*) + r - \mathsf{AE}_0(f^*) \right) V(L_{\max}) - R_{\max} \\
&= \left( \mathsf{AP}_0(0) + \mathsf{AP}_0(f^*) + r - \mathsf{AE}_0(f^*) \right) V(L_{\max}) - (\mathsf{AP}_0(0) + r) V(L_{\max}) \\
&= \left( \mathsf{AP}_0(f^*) - \mathsf{AE}_0(f^*) \right) V(L_{\max}) \\
&> 0.
\end{aligned}$$

Here, the first inequality follows from the suboptimality of $f^*$ for the am-AMM, and we also use the fact that $\Pi^{\mathsf{LP}}_{\mathsf{ff}}(f^*, L_{\max}) = 0$ since it is an equilibrium for the ff-AMM. The last equality follows from Assumption 3.

Since the pool manager earns positive profits when the rent is $R_{\max}$, and equilibrium rent $R^*$ musty satisfy $R^* > R_{\max}$, therefore and equilibrium fee-auction AMM utility $L^* > L_{\max}$.

## B Withdrawal Fees

Here, we show that a withdrawal fee of $1 - \frac{2 \cdot \sqrt{f_{cap}}}{f_{cap}+1}$ is sufficient to protect managers from strategic liquidity withdrawals. With a fee cap of 1%, this comes out to approximately 0.00001238%, or about 0.1238 basis points — a $12.38 fee on a $1 million position.

Our goal is to set a withdrawal fee to prevent opportunistic withdrawals of liquidity (in response to arbitrage opportunities) from reducing manager profits. Under our assumptions, the manager's profit from a given arbitrage opportunity is capped once the price moves by $f_{cap}$, the maximum allowable swap fee. Any larger price move results in MEV for the block proposer (since any arbitrageur can capture the excess portion with a swap), rather than profit for the manager. Therefore, we only need to set a withdrawal fee that is greater than the value from arbing that liquidity in response to an increase of a factor of $f_{cap}$. (A decrease by a factor of $f_{cap}$ will result in a smaller arbitrage opportunity.)

For this proof, we use the same conventions as Section 3, where reserves are expressed in terms of two assets $x$ and $y$, asset $y$ is defined as the numéraire for all prices and valuations, and "liquidity" for a position is defined as $\sqrt{xy}$ where $x$ and $y$ are the position's reserves of assets $x$ and $y$, respectively.

Suppose without loss of generality that a liquidity provider is providing 1 unit of liquidity, and the current midpoint price on the AMM is $p_{amm}$. Since $p_{amm} = \frac{y_{now}}{x_{now}}$ and $x_{now} \cdot y_{now} = 1$, the liquidity provider's current reserves of assets X and Y are $x_{now} = \frac{1}{\sqrt{p_{amm}}}$ and $y_{now} = \sqrt{p_{amm}}$.

We suppose price has increased by $f_{cap}$, making the true price $p_{true} = f_{cap} \cdot p_{amm}$.

The current valuation of the liquidity is:

$$v_{now} = p_{true} \cdot x_{now} + y_{now} = f_{cap} \cdot p_{amm} \cdot \frac{1}{\sqrt{p_{amm}}} + \sqrt{p_{amm}} = (1 + f_{cap}) \cdot \sqrt{p_{amm}} \tag{8}$$

The valuation of the liquidity after the manager arbitrages the pool to $p_{true}$ ($f_{cap} \cdot p_{amm}$) would be:

$$v_{after} = f_{cap} \cdot p_{amm} \cdot \frac{1}{\sqrt{f_{cap} \cdot p_{amm}}} + \sqrt{f_{cap} \cdot p_{amm}} = 2 \cdot \sqrt{f_{cap} \cdot p_{amm}} \tag{9}$$

To prevent strategic withdrawal from being profitable for liquidity providers at the expense of managers, we can impose a withdrawal fee of $\frac{v_{now} - v_{after}}{v_{now}}$, which simplifies to:

$$f_{withdrawal} = \frac{v_{now} - v_{after}}{v_{now}} = 1 - \frac{2 \cdot \sqrt{f_{cap}}}{1 + f_{cap}} \tag{10}$$