Batching Trades on Automated Market Makers

Andrea Canidio[1]
CoW Protocol, Lisbon, Portugal

Robin Fritsch
Cow Protocol, Lisbon, Portugal
ETH Zürich, Switzerland

Abstract

We consider an automated market maker (AMM) in which all trades are batched and executed at a price equal to the marginal price (i.e., the price of an arbitrarily small trade) after the batch trades. We show that such an AMM is a *function maximizing* AMM (or FM-AMM): for given prices, it trades to reach the highest possible value of a given function. Competition between arbitrageurs guarantees that an FM-AMM always trades at a fair, equilibrium price, and arbitrage profits (also known as LVR) are eliminated. Sandwich attacks are also eliminated because all trades occur at the exogenously-determined equilibrium price. Finally, we show that our results are robust to the case where the batch has exclusive access to the FM-AMM, but can also trade on a traditional constant function AMM.

2012 ACM Subject Classification Applied computing → Economics

Keywords and phrases Arbitrage profits, Loss-vs-Rebalancing (LVR), MEV, Sandwich attacks, AMM, Mechanism design, Batch trading

Digital Object Identifier 10.4230/LIPIcs.AFT.2023.24

Related Version Details missing from the text are available in the full version of the paper.
Full Version: `https://arxiv.org/abs/2307.02074` [5]

Acknowledgements We are grateful to Felix Leupold and Martin Köppelmann for initial discussions on batch trading on AMM that led to the writing of this paper. We also thank Haris Angelidakis, Eric Budish, Agostino Capponi, Felix Henneke, Fernando Martinelli, Ciamac Moallemi, Andreas Park, and Anthony Lee Zhang for numerous comments and suggestions.

1 Introduction

This design of Constant Function Automated Market Makers (CFAMMs) has two well-recognized flaws. First, liquidity providers (LPs) trade at a loss whenever there is a rebalancing event. More precisely, when the underlying value of the assets changes, the first informed arbitrageur who trades with the CFAMM earns a profit by aligning the CFAMM price with the new equilibrium price. These profits are at the expense of LPs, who suffer a "loss-vs-rebalancing" (LVR) [16]. Second, traders are routinely exploited by attackers, most commonly via sandwich attacks in which an attacker front-runs a victim's swap with the same swap and then back-runs it with the opposite swap. Doing so allows the attacker to "buy cheap" and "sell expensive" while forcing the victim to trade at less favorable terms.

This paper proposes a novel AMM design that avoids both problems. In our design, all trades that reach the AMM during a period are batched together and executed at a price equal to the new marginal price on the AMM – that is, the price of executing an arbitrarily small trade after the batch trades. We derive the trading function of such an AMM and show two interesting equivalences. First, this AMM is *function maximizing* because, for given

[1] corresponding author

prices, it maximizes the value of a given function subject to a budget constraint. For this reason, we call our design a function-maximizing AMM, or *FM-AMM*. Also, if the function is a standard Cobb-Duglas objective function (i.e., the weighted sum of two natural logs), then for given prices, the FM-AMM LPs run a passive investment strategy: absent trading fees, the total value of the two reserve assets is shared between the two assets according to some pre-specified weights. Finally, we show that an FM-AMM does not satisfy path independence: traders can obtain a better price by splitting their trades into smaller orders, which is why batching is required.

Our main contribution is to consider the behavior of such an AMM in the presence of arbitrageurs, who have private information relative to the equilibrium prices (determined, for example, on some very liquid off-chain location). Competition between arbitrageurs guarantees that the batch always trades at the equilibrium price, and arbitrage profits are eliminated. Intuitively, if this were not the case, some arbitrageurs would want to trade with the batch and, by doing so, would push the price on the batch in line with the equilibrium. This also eliminates all forms of MEV extraction, such as, for example, sandwich attacks: arbitrageurs will always act so to remove deviations from the equilibrium price, therefore making it impossible to manipulate the FM-AMM price. The benefit of contributing liquidity to an FM-AMM relative to a traditional CFAMM is that FM-AMM LPs earn the arbitrage profits generated by rebalancing the CFAMMs. Because these arbitrage profits are larger for more volatile prices (as they lead to more frequent and larger rebalancing, see [16] and [15]), holding everything else equal, the benefit of providing liquidity to an FM-AMM relative to a CFAMM increases with the price volatility.

We extend the theoretical analysis to include a traditional CFAMM, which is important because trading on an FM-AMM may be more expensive than trading on a traditional CFAMM. We introduce the role of the batch operator, who acts in the traders' interest by splitting the batch between the FM-AMM and the traditional CPAMM. We consider the case in which there is a liquid, external trading venue where the equilibrium price is determined and the case in which the entire market is composed of the FM-AMM and the traditional CFAMM. Our results are robust to this extension: we show that arbitrageurs will trade with the batch and on the regular AMM to keep their prices in line with each other (and with the externally-defined equilibrium price, if it exists). The only difference is that the optimal strategy for arbitrageurs may be to trade with the batch and then back-run the batch on the CFAMM. Doing so guarantees that the prices are in equilibrium, and also generates strictly positive profits for the arbitrageur who can successfully back-run the batch.

The remainder of the paper is organized as follows. We now discuss the relevant literature. In Section 2, we introduce the FM-AMM. In Section 3 we consider the behavior of FM-AMM in the equilibrium of a game with informed arbitrageurs and noise traders. Section 4 presents the extension with multiple trading venues. Section 5 discusses the empirical estimation of the returns of providing liquidity to a zero-fee FM-AMM contained in the full version of the paper [5]. The last section concludes. All proofs and mathematical derivations missing from the text are in the appendix.

Relevant literature

The intuition for our main result is closely related to Budish et al. [3], who study the batching of trades in the context of traditional finance as a way to mitigate the high-frequency-trading (HFT) arms race and protect regular (or slow) traders. The main result is that batching trades force informed arbitrageurs to compete in prices instead of speed because the priority of execution within the batch is given based on price. The intuition in our model is similar,

although we assume that arbitrageurs compete in quantities rather than in prices: if the price on an FM-AMM differs from the equilibrium price, competing arbitrageurs will submit additional trades to exploit the available arbitrage opportunity, but by doing so, they push the price on the FM-AMM in line with the equilibrium.

Several authors argued that AMM's design allows arbitrageurs to profit at the expense of LPs. Aoyagi [1], Capponi and Jia [6], and Milionis et al. [16] provide theoretical models that illustrate this possibility. In particular, they consider a continuous time model with zero fees and derive a closed-form formula to measure LPs returns and the cost they face when trading with informed arbitrageurs (which they call loss-vs-rebalancing or LVR). Milionis et al. [15] extend this analysis to the case of discrete-time and strictly positive trading fees. They use the term *arbitrageur profits* to indicate LPs losses, a term we adopt because both our model and our empirical analysis are in discrete time and have fees. Aoyagi [1] and Capponi and Jia [6] draw the implication of this cost for liquidity provision.

A second important limitation of CFAMMs is that they enable sandwich attacks (see [17]). These attacks are quantitatively relevant. For example, Torres et al. [21] collected on-chain data from the inception of Ethereum (July 30, 2015) until November 21, 2020 and estimated that sandwich attacks generated 13.9M USD in profits. Qin et al. [18] consider a later period (from the 1st of December 2018 to the 5th of August 2021) and find that sandwich attacks generated 174.34M USD in profits. Our design eliminates these attacks. We are therefore related to the growing literature proposing mechanisms to prevent malicious re-ordering of transactions (of which sandwich attacks are an example), especially those that can be implemented at the smart-contract level [2, 10, 4, 9][2].

Several initial discussions on designing "surplus maximizing" or "surplus capturing" AMMs occurred informally on blog and forum posts (see Leupold [14], Josojo [12], and Della Penna [8]). Goyal et al. [19] provide an axiomatic derivation of the surplus-maximizing AMM. Relative to their work, our contribution is to place this new type of AMM in a context with arbitrageurs and other trading venues. Schlegel and Mamageishvili [20] also study AMM from an axiomatic viewpoint. In particular, they discuss path independence, which FM-AMMs violate.

We conclude by noting that an FM-AMM is also an oracle: it exploits competition between arbitrageurs to reveal on-chain the price at which these arbitrageurs can trade off-chain. It is, therefore, related to the problem of Oracle design (as discussed, for example, by Chainlink [7]).

2 The function-maximizing AMM

In this section, we introduce the main concepts of interest using a simple constant-product function (both for the CFAMM and the FM-AMM), no fees, and keeping formalities to the minimum. The definitions are generalized at the end of the section. We refer the reader to the paper's conclusions for a discussion of additional design choices, and to the main version of the paper [5] for a generalization of our results.

As a preliminary step, we derive the trading function of a constant product AMM, the simplest and most common type of CFAMM. Suppose that there are only two tokens, ETH and DAI. A constant-product AMM (CPAMM) is willing to trade as long as the product of

[2] Another strand of the literature studies how to prevent malicious re-ordering of transactions by modifying the infrastructure that underpins how transactions are sent. See, for example, [13] and the literature review in [11].

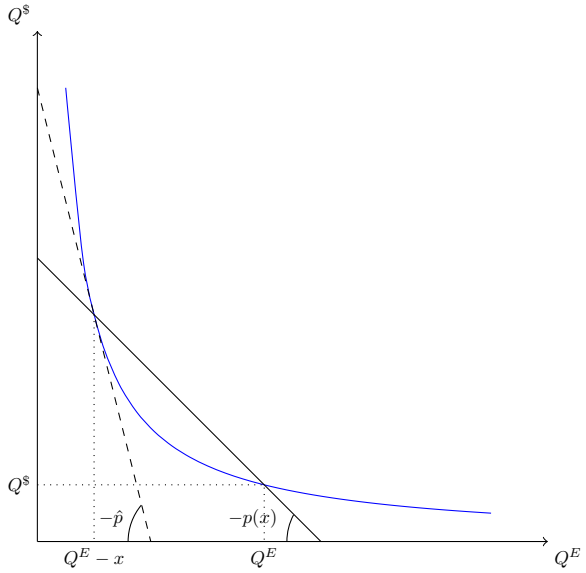

Figure 1 Initially, the liquidity reserves of the CPAMM are $Q^\$$ and Q^E. A trader then purchases x ETH at an average price $p(x)$. Note that, after the trade, the marginal price on the CPAMM (that is, the price for an arbitrarily small trade) is $\hat{p} \neq p(x)$.

its liquidity reserves remains constant (see Figure 1 for an illustration). Call $Q^\$$ and Q^E its initial liquidity reserves in DAI and ETH, respectively, and $p^{CPAMM}(x)$ the average price at which the CPAMM is willing to trade x ETH, where $x > 0$ means that CPAMM is selling ETH while $x < 0$ means that the CPAMM is buying ETH. For the product of the liquidity reserves to be constant, it must be that

$$Q^\$ \cdot Q^E = (Q^\$ + p^{CPAMM}(x)x)(Q^E - x)$$

or

$$p^{CPAMM}(x) = \frac{Q^\$}{Q^E - x}.$$

Note that the marginal price of a CPAMM (i.e., the price to trade an arbitrarily small amount) is given by the ratio of the two liquidity reserves. The key observation is that, in a CPAMM, a trader willing to trade x pays a price that is different from the marginal price after the trade. This is precisely the reason why arbitrageurs can exploit a CPAMM: an arbitrageur who trades with the CPAMM to bring its marginal price in line with some exogenously-determined equilibrium price does so at an advantageous price (and hence makes a profit at the expense of the CPAMM).

Instead, in the introduction, we defined an FM-AMM as an AMM in which, for every trade, the average price equals the marginal price after the trade (we may call this a *clearing-price-consistent AMM*). For ease of comparison with the CPAMM described earlier, suppose that the FM-AMM function is the product of the two liquidity reserves, and hence, that its marginal price is the ratio of its liquidity reserves. The price function $p(x)$ for buying x ETH on the FM-AMM is implicitly defined as

$$p(x) = \frac{Q^\$ + x \cdot p(x)}{Q^E - x},$$

where the RHS of the above expression is the ratio of the two liquidity reserves after the trade. Solving for $p(x)$ yields:

$$p^{FM-AMM}(x) \equiv p(x) = \frac{Q^{\$}}{Q^E - 2x},$$

which implies that the FM-AMM marginal price is, indeed, the ratio of the liquidity reserves. Hence, a given trade on the FM-AMM generates twice the price impact than the same trade on the traditional CPAMM (cf. the expression for $p^{CPAMM}(x)$).

Interestingly, an FM-AMM can also be seen as a price-taking agent maximizing an objective function. If its objective function is the product of the two liquidity reserves, then for a given price p the FM-AMM supplies x ETH by solving the following problem:

$$x^{FM-AMM}(p) = \text{argmax}_x \left\{ (Q^E - x)(Q^{\$} + p \cdot x) \right\}.$$

It is easy to check that the FM-AMM supply function is:

$$x^{FM-AMM}(p) = \frac{1}{2}\left(Q^E - \frac{Q^{\$}}{p} \right).$$

Hence, to purchase x ETH on the FM-AMM, the price needs to be, again:

$$p^{FM-AMM}(x) = \frac{Q^{\$}}{Q^E - 2x}.$$

It follows that, whereas a traditional CPAMM always trades along the same curve given by $Q^{\$}Q^E$, the FM-AMM trades as to be on the highest possible curve. With some approximation, we can therefore see an FM-AMM as a traditional CPAMM in which additional liquidity is added with each trade. See Figure 2 for an illustration.

A final observation is that the FM-AMM's trading function is equivalent to

$$p \cdot (Q^E - x^{FM-AMM}(p)) = Q^{\$} + p \cdot x^{FM-AMM}(p).$$

In other words, for a given p, the values of the two liquidity reserves are equal after the trade. Therefore, the FM-AMM is trading to implement a passive investment strategy, in which the total value of the two reserves is equally split between the two assets (that is, a passive investment strategy with weights 1/2, 1/2). It is easy to check that the FM-AMM can implement any passive investment strategy with fixed weights $(\alpha, 1-\alpha)$ by specifying the objective function as $(Q^E)^{\alpha}(Q^{\$})^{1-\alpha}$ for an appropriate $\alpha \in (0,1)$.

2.1 Path-dependence (or why batching trades is necessary)

CFAMMs (without fees) are path-*independent*: splitting a trade into multiple parts and executing them sequentially does not change the average price of the trade. This property does not hold for an FM-AMM because traders can get better prices by splitting their trade. In fact, they can get approximately the same price as on the corresponding CFAMM by splitting their trade into arbitrarily small parts. This is why an FM-AMM's trading function can be implemented only if trades are batched.

To see this, note that a trade on the FM-AMM with product function changes the reserves as follows:

$$\left(Q^{\$}, Q^E\right) \quad \rightarrow \quad \left(Q^{\$}\left(\frac{Q^E - x}{Q^E - 2x}\right), Q^E - x \right)$$

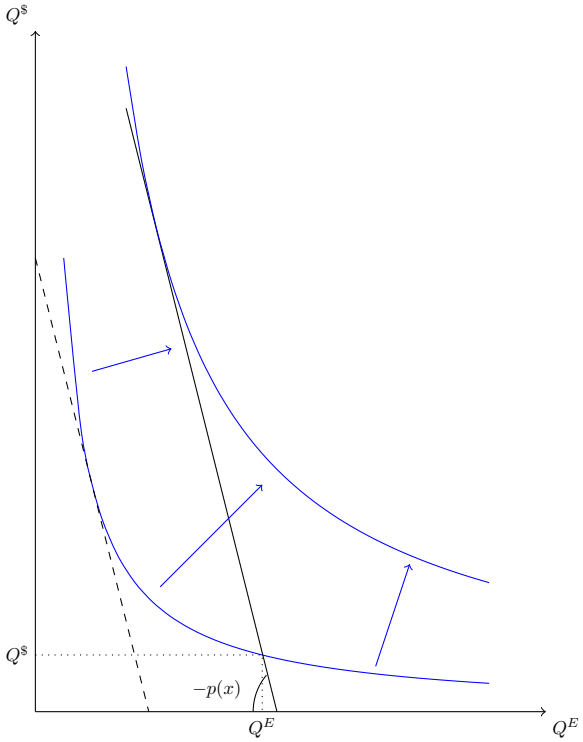

Figure 2 On an FM-AMM, the price at which a given trade x is executed equals the marginal price after the trade is executed. This implies that an FM-AMM "moves up" the curve with each trade.

By instead splitting the trade into smaller parts $\sum_{i=1}^{n} x_i = x$ and executing them sequentially, the reserves of the FM-AMM will change to

$$\left(Q^\$ \prod_{i=1}^{n} \frac{(Q^E - \sum_{j=1}^{i-1} x_j) - x_i}{(Q^E - \sum_{j=1}^{i-1} x_j) - 2x_i}, Q^E - \sum_{i=1}^{n} x_i \right).$$

Setting $x_i = \frac{1}{n}x$ and letting $n \to \infty$ leads to the DAI reserves after the trade being

$$\lim_{n\to\infty} Q^\$ \frac{Q^E - \frac{1}{n}x}{Q^E - \frac{n+1}{n}x} = Q^\$ \frac{Q^E}{Q^E - x}$$

since the product becomes a telescoping product and collapses. This term exactly equals the DAI reserve of a CPAMM after these trades. Hence, to have an FM-AMM, it is necessary to prevent splitting orders by imposing the batching of trades.

2.2 Generalization of definitions

We now generalize our definitions. First of all, an CFAMM is an entity that accepts or rejects trades based on a pre-set rule. Such rule can be derived from the CFAMMs liquidity reserves $(Q^\$, Q^E) \in \mathbb{R}^2_+$ and the CFAMM function $\Psi : \mathbb{R}^2_+ \to \mathbb{R}$. We assume that the CFAMM function is continuous, it is such that $\Psi(Q^\$, 0) = \Psi(0, Q^E) = \Psi(0, 0)$ for all $Q^\$$ Q^E, that it is strictly increasing in both its arguments whenever $Q^\$ > 0$ and $Q^E > 0$, and that it is strictly quasiconcave. The difference between different types of CFAMMs is how the function $\Psi(.,.)$ and the liquidity reserves $(Q^\$, Q^E)$ determine what trades will be accepted and rejected by the CFAMM.

▶ **Definition 1** (Constant Function Automated Market Maker). *For given liquidity reserves* $(Q^{\$}, Q^{E})$ *and function* $\Psi : \mathbb{R}^2_+ \to \mathbb{R}$*, a constant function automated market maker (CFAMM) is willing to trade* x *for* $y = p(x)x$ *if and only if*

$$\Psi(Q^{\$} + p(x)x, Q^{E} - x) = \Psi(Q^{\$}, Q^{E}).$$

Our first goal is to define an AMM that is *clearing-price-consistent* in the sense that, for every trade, the average price of the trade equals the marginal price after the trade. This requirement, together with the specification of a marginal price, already fully defines the AMM. In particular, a clearing-price-consistent AMM can be defined to match the marginal price for an arbitrary CFAMM function Ψ.

▶ **Definition 2** (Clearing-Price Consistent AMM). *For given liquidity reserves* $(Q^{\$}, Q^{E})$ *and function* $\Psi : \mathbb{R}^2_+ \to \mathbb{R}$*, let*

$$p_{\Psi}^{margin}(Q^{\$}, Q^{E}) = \frac{\frac{\partial \Psi(Q^{\$}, Q^{E})}{\partial Q^{E}}}{\frac{\partial \Psi(Q^{\$}, Q^{E})}{\partial Q^{\$}}}$$

be the marginal price of the AMM for reserves $Q^{\$}, Q^{E}$*. A clearing-price consistent AMM is willing to trade* x *for* $y = p(x)x$ *if and only if*

$$p(x) = p_{\Psi}^{margin}\left(Q^{\$} + p(x)x, Q^{E} - x\right). \tag{1}$$

Note that, given our assumptions on $\Psi(.,.)$, whenever $Q^{\$} > 0$ and $Q^{E} > 0$, the marginal price is strictly increasing in the first argument and strictly decreasing in the second argument, converges to zero as $Q^{E} \to \infty$ or $Q^{\$} \to 0$, and to infinity as $Q^{E} \to 0$ or $Q^{\$} \to \infty$.

Second, we define a *function-maximizing AMM (FM-AMM)* that maximizes the objective function Ψ instead of keeping it constant:

▶ **Definition 3** (Function-Maximizing AMM). *For given liquidity reserves* $(Q^{\$}, Q^{E})$ *and function* $\Psi : \mathbb{R}^2_+ \to \mathbb{R}$*, a function-maximizing AMM is willing to trade* x *for* $y = p(x) \cdot x$ *if and only if* $p(x) = x^{-1}(p)$*, where*

$$x(p) := argmax_x \left\{ \Psi\left(Q^{\$} + p \cdot x, Q^{E} - x\right) \right\}. \tag{2}$$

The next proposition establishes the equivalence between clearing-price-consistent and function-maximizing AMMs.

▶ **Proposition 4.** *For given liquidity reserves* $(Q^{\$}, Q^{E})$ *and function* $\Psi : \mathbb{R}^2_+ \to \mathbb{R}$*, an AMM is function maximizing if and only if it is clearing-price consistent.*

Proof. Under our assumptions, solving (2) is equivalent to satisfying the first-order condition, which is equivalent to (1). ◀

3 The model

Equipped with the full description of an FM-AMM, we can now study its behavior in an environment with traders and arbitrageurs. We limit our analysis to the product function.

The game comprises n noise traders, each wanting to trade a_i units of ETH. We adopt the convention that if $a_i > 0$ trader i wants to buy ETH, while if $a_i < 0$ trader i wants to sell ETH. Call $A = \sum_i a_i$ the aggregate demand for ETH from noise traders, assumed small

relative to the FM-AMM liquidity reserves Q^E and $Q^\$$.[3] Next to traders, a large number of cash-abundant, competing arbitrageurs, who can trade as part of the batch and on some external trading venue, assumed much larger and more liquid than the combination of our n traders and the FM-AMM. The equilibrium price for ETH on this external trading venue is p^* and is unaffected by trades on the FM-AMM. Arbitrageurs aim to profit from price differences between the FM-AMM and the external trading venues. Arbitrage opportunities will be intertemporal (over short intervals). Hence, for ease of derivations, we assume that arbitrageurs do not discount the future.

The timing of the game is discrete. Even periods are when on-chain transactions occur. Even periods are, therefore, better interpreted as different blocks. Odd periods are when off-chain events occur. In these periods, first, the equilibrium price p^* may change, and then traders/arbitrageurs can submit trades for inclusion in the batch.

We are now ready to derive our main proposition.

▶ **Proposition 5.** *Suppose that, at the end of an even period, the reserves of the FM-AMM are Q^E and $Q^\$$. In the equilibrium of the subsequent odd period, after p^* is realized, noise traders will collectively submit trade A to the batch, and arbitrageurs will collectively submit trade $y(p^*)$ such that $p^{FM-AMM}(A + y(p^*)) = p^*$. Hence, in equilibrium, the FM-AMM price is always equal to p^*.*

The key intuition is that all arbitrageurs can submit trades on the batch. Hence, there is no equilibrium in which there is an exploitable arbitrage opportunity; otherwise, some arbitrageurs will want to submit additional trades on the batch. Arbitrage opportunities are absent whenever $p^{FM-AMM}(A + y(p^*)) = p^*$, which is the unique equilibrium. Hence, competition between arbitrageurs guarantees that the price at which the AMM trades is always correct, even if the AMM has no way of directly observing such price. Unlike a traditional CPAMM, arbitrageurs earn zero by rebalancing an FM-AMM.

4 Extension: multiple trading venues

We modify our theoretical model by introducing a traditional CPAMM. This is a relevant extension because, as already discussed, trading on FM-AMM may be more expensive than trading on a traditional CPAMM, in the sense that for given liquidity pools, a given trade on a CF-AMM has twice the price impact than the same trade on a CPAMM. Naive noise traders may therefore ignore the equilibrium effects and prefer to trade on the CPAMM.

To address this concern, we introduce the figure of the batch operator as the sole agent who can access the FM-AMM. The batch operator has two roles:[4]

- It enforces the batching of trades, including uniform prices for all trades on the batch.
- Conditional on batching trades, the batch operator acts in the traders' interest. In particular, the batch operator will optimally split a trade between the FM-AMM and the traditional CPAMM to obtain the best possible price for the traders in the batch

[3] In practice, we assume that trading A on the FM-AMM has a negligible price impact. Else, we should treat orders from noise traders as limit orders. In this case, all our results continue to hold at the cost of additional notation.

[4] The batch operator is modeled around CoW Protocol (`www.cow.fi`). CoW Protocol collects intentions to trade off-chain, which are executed as a batch by accessing multiple trading venues. Cow Protocol enforces uniform clearing prices so that all traders in the same batch face the same prices.

The rest of the game is similar to the one discussed earlier, except that the large trading venue in which the equilibrium price is determined may or may not exist. If it exists, we say that the market is deep. Else, the only trading venues are the FM-AMM and the CPAMM, and we say that the market is shallow.[5]

Again, there are noise traders and arbitrageurs. Arbitrageurs trade so to exploit price differences between the trading venues that they can access. Even periods of the game are when on-chain transactions occur and are better interpreted as different blocks. Odd periods are when off-chain events occur. In these periods, traders/arbitrageurs can submit trades for inclusion in a batch. Other events may also happen, depending on whether the market for trading ETH against DAI is deep or shallow. For simplicity, both FM-AMM and the CPAMM charge zero fees.

We start by deriving the optimal behavior of the batch operators

4.1 The batch operator optimal behavior

The first step is to derive the optimal behavior of the batch operator, who can trade both on an FM-AMM (with reserves $Q^{\$,FM-AMM}$ and $Q^{E,FM-AMM}$) and on a CPAMM (with reserves $Q^{\$,CPAMM}$ and $Q^{E,CPAMM}$). If the batch needs to purchase X, it will split the trade between x^{CPAMM} and $x^{FM-AMM} = X - x^{CPAMM}$ so as to minimize the total expenditure, that is

$$\min_{x^{CPAMM}} \left\{ x^{CPAMM} \frac{Q^{\$,CPAMM}}{Q^{E,CPAMM} - x^{CPAMM}} + x^{FM-AMM} \frac{Q^{\$,FM-AMM}}{Q^{E,FM-AMM} - 2x^{FM-AMM}} \right\}$$

There is a special case in which the math simplifies: that of equal reserves in the two AMMs. In this case, the solution to the above minimization problem is $x^{CPAMM} = \frac{2}{3}X$; because the price impact on the FM-AMM is twice that of the traditional CPAMM, the batch will optimally route a given trade 1/3 on the FM-AMM and 2/3 on the standard CPAMM. As a result, the (average) price that the batch pays on the CPAMM equals the price the batch pays on the FM-AMM. We illustrate this case in Figure 3.

To simplify the analysis, we, therefore, assume that, initially, $Q^{\$,CPAMM} = Q^{\$,FM-AMM}$ and $Q^{E,CPAMM} = Q^{E,FM-AMM}$.

4.2 Deep market

We say that the market is deep whenever there is an exogenously-determined equilibrium price for ETH, which we call p^*. In this case, during odd periods (off-chain), first, the equilibrium price p^* may change, and then traders/arbitrageurs can submit trades for inclusion in the batch.

Suppose that, initially, the equilibrium price is p_0^*, and both AMMs are in equilibrium so that $p_0^* = \frac{Q^\$}{Q^E}$. In a given odd period, the equilibrium price then jumps to $p_1^* \neq p_0^*$, and stays constant afterward. We study the adjustment to the new equilibrium depending on whether the batch or an arbitrageur is first in the block.[6]

[5] Of course, the ETH/DAI market is deep. The point is to generalize the example to token pairs that may not be.

[6] In deriving the equilibrium, we assume that a player observes on-chain transactions included earlier in a block and can use this observation to craft their transaction. This is a shorthand for a transaction that specifies a strategy to follow as a function of the possible on-chain behavior of the preceding players.

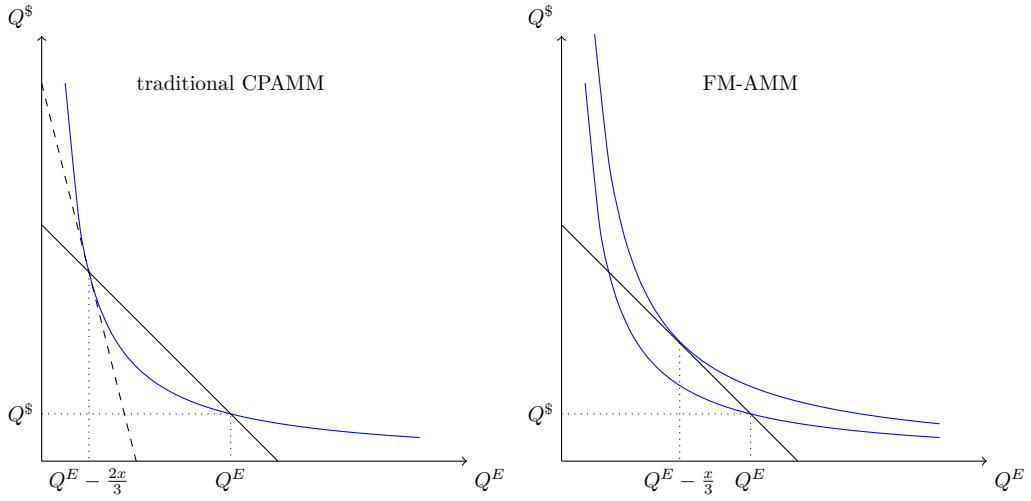

Figure 3 Starting from the initial reserves (assumed equal in the two AMMs) the batch trades so that the new marginal price on the FM-AMM equals the average price on the traditional CPAMM (represented by the two solid black lines). Note that the marginal price on the traditional CPAMM (dashed line) differs from the average price at which the batch trades.

The batch is at the top of the block

In this case, upon observing the new price, arbitrageurs submit trades for inclusion in the batch. Competition among arbitrageurs guarantees that their total trade on the batch is such that the price on the batch is p_1^*. This implies that the price on the FM-AMM is exactly p_1^*. However, if the batch traded on the traditional CPAMM at an *average* price of p_1^*, then the new *marginal* price on the traditional CPAMM will be different from p_1^*. More precisely, call $x^{CPAMM,*}$ the amount of ETH exchanged by the batch on the traditional CPAMM, defined as

$$x^{CPAMM,*}: \; p_1^* = \frac{Q^\$}{Q^E - x^{CPAMM,*}}$$

After such trade, the marginal price on the traditional CPAMM is given by:

$$\frac{Q^\$ + p_1^* x^{CPAMM,*}}{Q^E - x^{CPAMM,*}} = \frac{p_1^* Q^E}{\frac{Q^\$}{p_1^*}} = \frac{(p_1^*)^2}{p_0^*} \neq p_1^*$$

where $Q^\$ + p_1^* x^{CPAMM,*} = p_1^* Q^E$ and $Q^E - x^{CPAMM,*} = \frac{Q^\$}{p_1^*}$ are, respectively, the DAI and ETH reserves on the CPAMM after the batch settles its trade. Hence, if $p_1^* > p_0^*$, then the marginal price on the traditional CPAMM will be greater than p_1^*; if instead $p_1^* < p_0^*$, then the marginal price on the traditional CPAMM will be smaller than p_1^*. The arbitrageur trading after the batch will then perform a trade y given by:[7]

$$\max_y \left\{ y \left(p_1^* - \frac{p_1^* Q^E}{\frac{Q^\$}{p_1^*} - y} \right) \right\}$$

[7] Note that if $y > 0$, then the arbitrageur buys from the CPAMM to sell at the equilibrium price p_1^*. If instead $y < 0$, then the arbitrageur buys at the equilibrium price p_1^* to sell to the CPAMM.

with first-order conditions

$$\frac{p_1^* + \left(\frac{p_1^* Q^E}{\frac{Q^\$}{p_1^*} - y}\right) y}{\frac{Q^\$}{p_1^*} - y} = p_1^*$$

It is easy to check that the numerator of the above expression is the DAI reserves after the batch settles and after the arbitrageur trades y. Similarly, the denominator is the ETH reserves after the trades from the batch and the arbitrageurs. Therefore, the arbitrageur's optimal trade re-aligns the marginal price on the CPAMM with the equilibrium price p_1^*.

An arbitrageur is at the top of a block

To start, note that, independently of what trades were submitted to the batch, the arbitrageur at the top of the block will exploit the arbitrage opportunity available on the traditional CPAMM, and trade on the CPAMM until its marginal price equals p_1^*. Hence, when arbitrageurs submit trades on the batch, they anticipate that the CPAMM price will be p_1^* by the time the batch trades. Competition guarantees that these arbitrageurs submit trades y to the batch until its price is also p_1^*, that is:

$$y: \quad p_1^* = \frac{Q^\$}{Q^E - 2(A+y)}$$

In this case, therefore, the batch does not trade on the traditional CPAMM, but only on the FM-AMM.

4.3 Shallow market

The market is shallow whenever the only venues to trade ETH against DAI are the FM-AMM and the traditional CPAMM. Hence, there is no externally-determined equilibrium price. The only arbitrage possibility is to exploit price differences between the FM-AMM and the traditional CPAMM.

Again, suppose that the two AMMs have equal reserves and, therefore, equal marginal prices. The source of the exogenous variation is the arrival of an aggregate trade A on the batch. After observing A, arbitrageurs can submit their trades to the batch. Again, there are two cases to consider, depending on who trades first.

The batch is at the top of the block

To start, note that if arbitrageurs were absent, then after the batch settles A (1/3 on the FM-AMM, the rest on the traditional CPAMM) the marginal prices of the two exchanges would be misaligned. There is therefore an arbitrage opportunity that the first arbitrageur can exploit, as the next proposition shows.

▶ **Proposition 6.** *In equilibrium, the first arbitrageur trading after the batch purchases*

$$y^* = 3\sqrt{2Q^E(2Q^E - A)} - 2(3Q^E - A) \tag{3}$$

with the batch and then sells the same amount on the CPAMM. The other arbitrageurs do not trade.

The proof of the proposition shows that y^* has the same sign as A. Hence, the first arbitrageur trades in the same direction as noise traders to move the price on the CPAMM even more, and then performs the opposite trade on the CPAMM. Furthermore, he can preempt the other arbitrageurs from exploiting the arbitrage opportunity. By doing so, he earns strictly positive profits.

The arbitrageur is at the top of the block

Also here, competition among arbitrageurs guarantees that, collectively, they trade to align the price of ETH on the batch with the marginal price of ETH on the CPAMM. However, unlike the previous case, here, both the first arbitrageur in the block and the first arbitrageur after the batch trade.

▶ **Proposition 7.** *In equilibrium, the first arbitrageur after the batch buys with the batch and then sells on the CPAMM. Again, his optimal trade is y^* given by* (3)*. Furthermore, the first arbitrageur purchases*

$$\sqrt{(Q^E)^2 - Q^E \frac{2}{3}(A + y^*)} - Q^E$$

on the CPAMM and then sells the same amount on the batch. The other arbitrageurs do not trade.

We previously discussed that y^* and A have the same sign. Hence, if for example $A > 0$, then the first arbitrageur purchases on the CPAMM (cheaply) and then sells on the batch (more expensive) to bring the CPAMM price in line with the price on the FM-AMM. If instead $A < 0$, the first arbitrageur sells on the CPAMM and then buys on the FM-AMM. The batch then trades both on the FM-AMM and the CPAMM, therefore moving the price on the CPAMM. Finally, similarly to the previous case, the first arbitrageur after the batch exploits the arbitrage opportunity by back-running the batch: he purchases with the batch and immediately sells on the CPAMM so to align the price on the FM-AMM with the marginal price on the CPAMM.

5 Discussion of empirical results

In the full version of the paper [5], we perform an empirical analysis to quantify the return of providing liquidity to a zero-fee FM-AMM, and compare them with the empirical returns of providing liquidity on Uniswap V3. We now briefly describe our method and report our main results.[8]

To simulate FM-AMM returns, we consider a counterfactual in which an FM-AMM existed from October 2022 to March 2023. We retrieve price data from Binance for several trading pairs and use the Proposition 5 to simulate an FM-AMM pool rebalanced to the Binance price in regular intervals (12 seconds, or 1 block). These rebalancing trades determine the evolution of the FM-AMM reserves and thereby the returns of its liquidity providers. Importantly, because we consider a zero-fee FM-AMM, its LPs earn no fees from noise traders. Equivalently, we could also assume that the FM-AMM does not receive any noise

[8] Note that in the full version of the paper, we consider how the return of providing liquidity of an FM-AMM changes with the fees charged by FM-AMM and with the frequency of rebalancing of the FM-AMM. Here we only report the results for a zero-fee FM-AMM that rebalances every block.

trading volume and is only rebalanced by arbitrageurs. Hence, the estimated LP returns should be considered a lower bound to the possible returns generated by an FM-AMM that also earns revenues from noise traders.

We then compare the returns of providing liquidity to our simulated FM-AMM to the empirical returns of providing liquidity to the corresponding Uniswap v3 pool. To calculate the return on a liquidity position in a Uniswap v3 pool, we consider three pools for which we also have Binance price data: WETH-USDT (with fee 0.05%), WBTC-USDT (with fee 0.3%), and WBTC-WETH (with fee 0.05%).[9] In each case, we simulate the return of a small full-range position. We then query the amount of fees the pool earned in a given block and the amount of liquidity that is "in range" at the end of the same block. [10] We then use the size of the simulated liquidity position in range to calculate the fees it earns.

Our method is based on two assumptions. First, we assume that the simulated liquidity position is too small to affect price slippage, the volume of trades, and the incentive to provide liquidity by other LPs. We also implicitly assume that the liquidity in the range is constant during a block. This last assumption introduces some non-systematic inaccuracies in our estimation. For example, if within a block, first some fees are collected and then some additional liquidity is introduced, our method attributes a fraction of these fees to the new liquidity even if, in reality, it did not earn any. If, instead, first some fees are collected and then some liquidity is withdrawn, our method does not attribute any of the fees to the liquidity that was withdrawn, while in reality, it did earn some fees. Similarly, if a price changes tick, our method shares all fees collected in a block among the liquidity available in the last tick, whereas, in reality, some of the fees are shared among the liquidity available at the initial tick.[11]

Note that our results do not depend on the size of the initial liquidity position. On Uniswap v3, a larger initial position earns proportionally more fees, but its ROI is the same. Similarly, on an FM-AMM, the size of the rebalancing trade scales proportionally with the available liquidity so that, again, its ROI is independent of its initial size. Also, for both Uniswap v3 and the FM-AMM, we consider a liquidity position that is non-concentrated (i.e. , a position over the entire price range $[0, \infty]$). If both positions are concentrated in the same (symmetrical) way, both Uniswap v3 fees and FM-AMM returns increase by the same factor as long as the price does not go out of range. So the comparison does not change, and the full-range comparison already constitutes a general comparison.

Our results are mixed: whether providing liquidity to our simulated FM-AMM generates higher returns than providing the same liquidity to Uniswap v3 depends on the token pair and the period we consider. However, these returns are similar: at the end of the 6-months period we consider, the differences in returns between an FM-AMM and Uniswap v3 across the three pools we study are -0.25% (for the ETH-USDT pool), 0.03% (for the BTC-USDT pool) and 0.11% (for the ETH-BTC pool). Also, during the period we consider, the maximum difference in value between the two liquidity positions (relative to their initial values) is 0.33% (for the ETH-USDT pool), 0.15% (for the BTC-USDT pool), and 0.12% (for the ETH-BTC pool). We conclude that the lowest bound on the return to providing liquidity to an FM-AMM is similar to the empirical return to providing liquidity to Uniswap v3.

---

[9] Note that, whereas most Binance prices are expressed in $USDT$, this stablecoin is not widely used in Uniswap v3: at the time of writing, if we exclude stablecoin-to-stablecoin pairs, the two pools we consider are the only pools with USDT in the top 30 Uniswap v3 pools.

[10] We use the Uniswap v3 subgraph to query the data. `https://thegraph.com/hosted-service/subgraph/uniswap/uniswap-v3`

[11] Besides being non-systematic, the inaccuracies introduced are likely to be extremely small. For the ETH-USDT pool, we calculate that the difference between assuming liquidity to be constant over *two* blocks instead of over one block is 0.0002% over 6 months. We expect the inaccuracies from assuming the liquidity to be constant over one block to be the same magnitude.

6 Conclusion

We conclude by discussing several design choices that are relevant to the implementation of an FM-AMM.

The first design choice is how to enforce the batching of trades. In the model, we assumed that the FM-AMM enforces batching by collecting intentions to trade off-chain and settling them on-chain at regular intervals, with all trades settled simultaneously facing the same prices. However, there could be other ways to enforce batching, for example, by leveraging proposer-builder separation (or PBS)[12]. In PBS, block builders (entities that assemble transactions in a block that are then forwarded to a proposer for inclusion in the blockchain) could compute the net trades that will reach the FM-AMM during that block. Builders will then include a message announcing this value at the beginning of the block, which the FM-AMM uses to compute the price at which all trades will be executed. If the proposer's announcement turns out to be correct at the end of the block, the FM-AMM will reward the builder (punishments can also be introduced if the block builder report is incorrect, see [14]).

A second design choice is the fee structure. An FM-AMM can charge fees in at least two ways: a fee could be charged for including a trade on the batch, and a separate one could be charged on the net trade that the batch settles on FM-AMM. The difference between the two fees is that some of the trades may be netted already on the batch without ever reaching the FM-AMM. See the full version [5] for an in-depth analysis of this problem.

Another important design choice for any AMM, that is omitted in this work, is what function to use, particularly its curvature: the degree to which the price changes with a given trade. Capponi and Jia [6] show that, in a traditional CPAMM, the choice of curvature is determined by a tradeoff. When the equilibrium price changes and the curvature is high, the CPAMM adjusts its price more rapidly, and LVR is low. At the same time, the amount LPs earn as trading fees is also low. On the other hand, if the curvature is low and the equilibrium price changes, then LVR is high. But because the CPAMM trades more, the amount earned by LPs as fees is also high. The optimal curvature balances these two elements and depends on the trade volume from noise traders vs. arbitrageurs. The fact that the FM-AMM always trades at the equilibrium price suggests that it should encourage trading by adopting a flatter curve relative to a traditional CPAMM. Formalizing this intuition is left for future work.

A final observation is that an FM-AMM eliminates adverse selection by creating competition among multiple informed arbitrageurs. However, if there is a single, informed trader, then this trader can still enjoy information rents and trade at an advantage against FM-AMM LPs.

References

1 Jun Aoyagi. Liquidity provision by automated market makers. *working paper*, 2020. URL: https://dx.doi.org/10.2139/ssrn.3674178.

2 Lorenz Breidenbach, Phil Daian, Florian Tramèr, and Ari Juels. Enter the hydra: Towards principled bug bounties and {Exploit-Resistant} smart contracts. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1335–1352, 2018.

3 Eric Budish, Peter Cramton, and John Shim. The high-frequency trading arms race: Frequent batch auctions as a market design response. *The Quarterly Journal of Economics*, 130(4):1547–1621, 2015.

[12] https://ethereum.org/en/roadmap/pbs/

4 Andrea Canidio and Vincent Danos. Commitment against front running attacks, 2023. arXiv:2301.13785.
5 Andrea Canidio and Robin Fritsch. Arbitrageurs' profits, lvr, and sandwich attacks: batch trading as an amm design response, 2023. arXiv:2307.02074.
6 Agostino Capponi and Ruizhe Jia. The adoption of blockchain-based decentralized exchanges, 2021. arXiv:2103.08842.
7 Chainlink. What is the blockchain oracle problem? Retrieved from https://chain.link/education-hub/oracle-problem on May 24, 2023, 2020. Online forum post.
8 Nicolás Della Penna. Mev minimizing amm (minmev amm). Retrieved from https://ethresear.ch/t/mev-minimizing-amm-minmev-amm/13775 on May 24, 2023, september 1 2022. Online forum post.
9 Matheus V. X. Ferreira and David C. Parkes. Credible decentralized exchange design via verifiable sequencing rules, 2023. arXiv:2209.15569.
10 Joshua S Gans and Richard T Holden. A solomonic solution to ownership disputes: An application to blockchain front-running. Technical report, National Bureau of Economic Research, 2022.
11 Lioba Heimbach and Roger Wattenhofer. Sok: Preventing transaction reordering manipulations in decentralized finance. In *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, AFT '22, pages 47–60, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3558535.3559784.
12 Josojo. Mev capturing amm (mcamm). Retrieved from https://ethresear.ch/t/mev-capturing-amm-mcamm/13336 on May 24, 2023, august 4 2022. Online forum post.
13 Mahimna Kelkar, Fan Zhang, Steven Goldfeder, and Ari Juels. Order-fairness for byzantine consensus. Cryptology ePrint Archive, Paper 2020/269, 2020. URL: https://eprint.iacr.org/2020/269.
14 F. Leupold. Cow native amms (aka surplus capturing amms with single price clearing). Retrieved from https://forum.cow.fi/t/cow-native-amms-aka-surplus-capturing-amms-with-single-price-clearing/1219/1 on May 24, 2023, november 1 2022. Online forum post.
15 Jason Milionis, Ciamac C. Moallemi, and Tim Roughgarden. Automated market making and arbitrage profits in the presence of fees, 2023. arXiv:2305.14604.
16 Jason Milionis, Ciamac C. Moallemi, Tim Roughgarden, and Anthony Lee Zhang. Automated market making and loss-versus-rebalancing, 2023. arXiv:2208.06046.
17 Andreas Park. Conceptual flaws of decentralized automated market making. Technical report, Working paper, University of Toronto, 2022.
18 Kaihua Qin, Liyi Zhou, and Arthur Gervais. Quantifying blockchain extractable value: How dark is the forest? In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 198–214. IEEE, 2022.
19 Geoffrey Ramseyer, Mohak Goyal, Ashish Goel, and David Mazières. Augmenting batch exchanges with constant function market makers, 2023. arXiv:2210.04929.
20 Jan Christoph Schlegel, Mateusz Kwaśnicki, and Akaki Mamageishvili. Axioms for constant function market makers, 2023. arXiv:2210.00048.
21 Christof Ferreira Torres, Ramiro Camino, et al. Frontrunner jones and the raiders of the dark forest: An empirical study of frontrunning on the ethereum blockchain. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1343–1359, 2021.

A Mathematical derivations

Proof of Proposition 5. The fact that $y(p^*)$ is the unique equilibrium is easily established by contradiction: suppose the equilibrium is $y' \neq y(p^*)$, which implies $p^{FM-AMM}(A+y') \neq p^*$. Then by the fact that $p^{FM-AMM}(A+y')$ is locally continuous, an arbitrageur could submit

an additional trade z such that $p^{FM-AMM}(A+y'+z) \neq p^*$ and earn strictly positive profits, which implies that y' is not an equilibrium. It is also easy to establish that $y(p^*)$ is an equilibrium, as no arbitrageur has any incentive to deviate. ◀

Proof of Proposition 6. Suppose that arbitrageurs submit an aggregate trade y on the batch and later $-y$ on the traditional CPAMM. The price on the batch is:

$$\frac{Q^\$}{Q^E - \frac{2}{3}(A+y)}$$

and the price on the reverse trade is

$$\frac{Q^\$ Q^E}{(Q^E - \frac{2}{3}(A+y))(Q^E - \frac{2}{3}(A+y)+y)}.$$

Competition between arbitrageurs guarantees that the marginal arbitrageur earns zero profits. This, in turn, implies that the price on the FM-AMM equals the marginal price on the CPAMM, that is

$$\frac{Q^\$}{Q^E - \frac{2}{3}(A+y)} = \frac{Q^\$ Q^E}{(Q^E - \frac{2}{3}(A+y)+y)^2}$$

with a unique solution in the range $[2A - 3Q^E, \frac{3}{2}Q^E - A]$ (so that both prices are strictly positive and well-defined), given by

$$y^* = 3\sqrt{2Q^E(2Q^E - A)} - 2(3Q^E - A).$$

Note that y^* is strictly increasing in A and is equal to zero whenever $A = 0$. This can be seen by verifying that the derivative with respect to A is positive for $A < \frac{7}{8}Q_E$ (which is true when the demand from the batch is somewhat smaller than the size of the pools, in particular in all realistic cases we are interested in). Hence, arbitrageurs trade on the batch in the same direction as noise traders.

We established that the arbitrageurs collectively purchase y^* ETH with the batch to sell these ETH on the CPAMM. We now complete the characterization of the equilibrium by showing that the first arbitrageur after the batch purchases the full amount.

Suppose not: there is an equilibrium in which n arbitrageurs collectively purchase $y^* = \sum_{i=1}^n y_i^*$ ETH, and the first arbitrageur after the batch purchases $y_1^* \neq y^*$ ETH. In this case, the first arbitrageur's profits are

$$\frac{y_1^* \cdot Q^\$}{Q^E - \frac{2}{3}(A+y^*)} - \frac{y_1^* \cdot Q^\$ Q^E}{(Q^E - \frac{2}{3}(A+y^*))(Q^E - \frac{2}{3}(A+y^*)+y_1^*)}$$

but because $y_1^* \neq y^*$, then

$$\frac{Q^\$}{Q^E - \frac{2}{3}(A+y)} \neq \frac{Q^\$ Q^E}{(Q^E - \frac{2}{3}(A+y)+y)^2}$$

which implies that $y_1^* \neq y^*$ does not maximize the first arbitrageur's profits, who therefore wants to deviate. ◀

Proof of Proposition 7. Call y_1 the amount purchased on the batch by the first arbitrageur, y_2 the amount purchased on the batch by all other arbitrageurs who trade before the batch, y_3 the amount purchased on the batch by the arbitrageur who trades immediately after the batch.

As a first step, we derive how the batch will trade $A + y_1 + y_2 + y_3$. Suppose the batch purchases x_1 on the traditional CPAMM and x_2 on the FM-AMM, with $A + y_1 + y_2 + y_3 = x_1 + x_2$. Total expenditure is

$$\frac{Q^{\$}Q^{E}x_1}{(Q^{E} + y_1 + y_2)(Q^{E} + y_1 + y_2 - x_1)} + \frac{Q^{\$}x_2}{Q^{E} - 2x_2}.$$

Which is minimized by trading

$$x_1 = \frac{2}{3}(A + y_3) + y_1 + y_2,$$

on the CPAMM and

$$x_2 = \frac{1}{3}(A + y_3),$$

on the FM-AMM. Intuitively, the batch operator first restores on the CPAMM the liquidity taken by the earlier arbitrageurs and then trades the rest of the batch 1/3 on the FM-AMM and the rest on the CPAMM.

Note that, to have an equilibrium, the sum of all trades must re-align the marginal price on the CPAMM with the price on the FM-AMM, which implies

$$\frac{Q^{\$}}{Q^{E} - \frac{2}{3}(A + y_3)} = \frac{Q^{\$}Q^{E}}{(Q^{E} - \frac{1}{3}(A - y_3))^2}.$$

The problem of the first arbitrageur after the batch is identical to that of the earlier case, because the batch has brought back the CPAMM to its initial state. He will perform the trade that re-aligns the FM-AMM price with that of the CPAMM, and is $y_3 = y^*$ as in Equation (3).

Consider then the problem from the point of view of the first arbitrageur. Because of the behavior of the first arbitrageur after the batch, the first arbitrageur in the block anticipates that the price on the FM-AMM does not depend on his trade. He therefore maximizes profits by buying tokens on the CPAMM until its marginal price is equal to the price on the FM-AMM (and on the batch), which is given by

$$\frac{Q^{\$}}{Q^{E} - \frac{2}{3}(A + y^*)} = \frac{Q^{\$}Q^{E}}{(Q^{E} + y_1^*)^2}$$

or

$$y_1^* = \sqrt{(Q^{E})^2 - Q^{E}\frac{2}{3}(A + y^*)} - Q^{E}$$

which is negative if $A > 0$ (i.e., the arbitrageur buys on the CPAMM and sells on the batch) and positive if $A < 0$ (i.e., the arbitrageur sells on the CPAMM and buys on the batch). ◀