Complexity – Approximation Trade-offs in Exchange Mechanisms: AMMs vs. LOBs

Jason Milionis[1][0000−0002−9460−9559], Ciamac C. Moallemi[2][0000−0002−4489−9260], and Tim Roughgarden[1,3][0000−0002−7163−8306]

[1] Department of Computer Science, Columbia University, New York NY 10027, USA
[2] Graduate School of Business, Columbia University, New York NY 10027, USA
[3] a16z Crypto, New York NY 10010, USA

Abstract. This paper presents a general framework for the design and analysis of exchange mechanisms between two assets that unifies and enables comparisons between the two dominant paradigms for exchange, constant function market markers (CFMMs) and limit order books (LOBs). In our framework, each liquidity provider (LP) submits to the exchange a downward-sloping demand curve, specifying the quantity of the risky asset it wishes to hold at each price; the exchange buys and sells the risky asset so as to satisfy the aggregate submitted demand. In general, such a mechanism is budget-balanced (i.e., it stays solvent and does not make or lose money) and enables price discovery (i.e., arbitrageurs are incentivized to trade until the exchange's price matches the external market price of the risky asset). Different exchange mechanisms correspond to different restrictions on the set of acceptable demand curves.

The primary goal of this paper is to formalize an approximation-complexity trade-off that pervades the design of exchange mechanisms. For example, CFMMs give up expressiveness in favor of simplicity: the aggregate demand curve of the LPs can be described using constant space (the liquidity parameter), but most demand curves cannot be well approximated by any function in the corresponding single-dimensional family. LOBs, intuitively, make the opposite trade-off: any downward-slowing demand curve can be well approximated by a collection of limit orders, but the space needed to describe the state of a LOB can be large.

This paper introduces a general measure of *exchange complexity*, defined by the minimal set of basis functions that generate, through their conical hull, all of the demand functions allowed by an exchange. With this complexity measure in place, we investigate the design of *optimally expressive* exchange mechanisms, meaning the lowest complexity mechanisms that allow for arbitrary downward-sloping demand curves to be approximated to within a given level of precision. Our results quantify the fundamental trade-off between simplicity and expressivity in exchange mechanisms.

As a case study, we interpret the widely-used Uniswap v3 AMM through the lens of our framework and trade-off results. We show that, under reasonable assumptions and in a precise sense, the Uniswap v3 design achieves a near-optimal compromise between complexity and approximation.

Keywords: Blockchain · Decentralized Finance · Automated Market Makers.

1 Introduction

Decentralized exchanges are now an integral part of the broader ecosystem of blockchains, as evidenced by their ever growing volume of transactions [24]. On model centralized exchanges, the exchange of a risky asset for a numéraire is typically carried out by an exchange mechanism known as an electronic limit order book (LOB), in which market participants specify quantities of shares of the risky asset they would like to trade at specified prices. Trades then occur as orders are matched in a greedy way: whenever there is overlap between bid and ask prices (i.e., between a buy and a sell), a trade is executed, and the matched orders are cleared from the LOB. LOBs therefore maintain and update a list of all the currently outstanding buy and sell orders.

LOBs face two types of challenges in an decentralized environment such as the Ethereum blockchain. First, because storage and computation in such an environment tend to be so scarce, implementing an LOB can be prohibitively expensive. Second, LOBs are well known suffer from liquidity problems in thin markets (markets with few buyers or sellers), for example, for "long-tail" crypto assets.

These challenges have motivated an alternative exchange design that has become very widely used in blockchains: automated market makers (AMMs) and, in particular, constant function market makers (CFMMs). Uniswap [1, 2] is the most well known and widely used example of a CFMM.

AMMs address the second challenge above by offering guaranteed liquidity, meaning at all times there is a spot price between 0 and ∞ at which the AMM is willing to buy or sell. AMMs like Uniswap address the first challenge by using only simple calculations and data structures. For example, for the canonical ("$xy = k$") constant product market maker, the state of mechanism can be described by two numbers (the quantities x and y held by the pool), and there is a simple closed-form formula (requiring only a small number of additions, multiplications, divisions, and square roots) for computing the quantity of the risky asset received in exchange for a specified amount of the numéraire (as a function of x and y).

In this paper, we provide a general framework for describing and reasoning about exchange mechanisms, which enables "apples-to-apples" comparisons between LOBs and AMMs on metrics such as complexity and expressiveness. More specifically, our contributions can be delineated as follows:

1. We provide a **common framework** for describing exchange mechanisms that encompasses both CFMMs and LOBs. In our general model, liquidity providers (LPs) submit to the exchange their preferences (in the form of what we define as **demand curves** for the risky asset) along with appropriate deposits of the risky asset and numéraire (see Section 2 for details).
2. We formalize the sense in which some methods of exchange are simpler than others, introducing a general notion of **exchange complexity**. Exchange

complexity is defined by the minimal set of basis functions that generate, through their conical hull, all of the demand functions allowed by an exchange. We classify the complexity of all the prominent types of exchange mechanisms (see Section 3 for details).

3. We characterize the **fundamental trade-off** between the *complexity* of an exchange (in a sense that we define) and the *expressibility* of an exchange as measured by its ability to approximate arbitrary preferences of the LPs (i.e., arbitrary demand curves). In particular, we prove matching (up to constant factors) upper and lower bounds on the minimum exchange complexity necessary to attain a specified approximation error (see Section 4 for details).
4. As a case study, we identify reasonable assumptions under which the widely-used Uniswap v3 exchange matches (up to constant factors) the best-possible complexity-approximation trade-off (see Section 5 for details).

1.1 Literature Review

The use of AMMs for decentralized exchange mechanisms was first proposed by Buterin [12] and Lu and Köppelmann [27]. The latter authors suggested a constant product market maker, which was first analyzed by Angeris et al. [7]. Angeris et al. [4, 5] define and use a reparameterization of a CFMM curve (established by Angeris and Chitra [3]) in terms of portfolio holdings of the pool with respect to the price as a tool to replicate payoffs and compute the pool's value function; we use this same reparameterization for different purposes, to define a general (i.e., not AMM-specific) framework of exchange and identify fundamental complexity-approximation trade-offs in exchange design.

A separate line of work seeks to design specific CFMMs with good properties by identifying good bonding functions, variations and combinations of CFMMs in a dynamic setting with a specific focus on optimizing fees, and minimizing arbitrage and slippage [6, 15–17, 19, 20, 23, 25, 31, 34, 35]. While fees could be easily integrated into our model, they have no bearing on complexity-approximation trade-offs and thus we generally ignore them in this paper for simplicity.

Some previous papers propose generalizations of CFMMs to somewhat wider classes of exchanges [11, 36] without considering LOBs.

CFMMs and LOBs have been compared before (in ways orthogonal to the questions studied here) [10, 13, 26]. Most of these works either compare the observed liquidities and the price efficiency of these mechanisms [13, 26] or study the same through the lens of arbitrage bounds [10]. Young [38] argues that AMMs can be interpreted as "smooth order books" and notes a type of non-uniform converse (with each possible state of a smooth order book represented using a different AMM). Chitra et al. [14] compare CFMMs and LOBs in terms of the number of arbitrage transactions necessary to recover from a liveness attack on the underlying blockchain.

Another line of work analyzes competition between CFMMs and LOBs and the consequent liquidity properties of both at equilibrium [8, 9, 13]. Goyal et al. [21] consider the computational complexity of computing such equilibria.

There is a large literature on the market microstructure of limit order books; see the textbook by O'Hara [30] and references therein. There are some examples of on-chain LOBs on high-throughput blockchains [28, 33].

Finally, Adams et al. [2] suggest that Uniswap v3's key feature is that "LPs can approximate any desired distribution of liquidity on the price space," with empirical backing provided by Huynh [22]; one application of our work is to put this intuition on sound mathematical footing. There is also work on Uniswap v3 from the LP perspective, such as how beliefs about future prices should guide the choice of an LP's demand curve [18, 29, 37].

2 Model

2.1 Model Primitives

We begin by describing our framework for exchange design. While this paper uses this framework specifically to study fundamental complexity-approximation trade-offs in exchange mechanisms, we believe it can serve also as a starting point for many future investigations.

Suppose there are two assets, a risky asset and a numéraire asset. Each LP comes separately to the exchange, and declares the amount of risky asset they would like to hold at each possible price p, i.e., a non-increasing, non-negative function $g_i \colon (0, \infty) \to \mathbb{R}^+$. We call the function $g_i(\cdot)$ the ith LP's **demand curve** for the risky asset, because it refers to the demand of the LP for the risky asset (i.e., we are considering the perspective of the LP). Assuming that the current price is p_0, the LP simultaneously deposits a quantity $g_i(p_0)$ of the risky asset in the common pool, along with an amount of numéraire given by the Riemann–Stieltjes integral

$$-\int_0^{p_0} p \, dg_i(p) \,. \tag{1}$$

Note that this integral is well-defined (though possibly infinite) since $g_i(\cdot)$ is monotonic. Moreover, the integral is non-negative since $g_i(\cdot)$ is non-increasing. In cases where $g_i(p)$ is differentiable, the differential takes the form $dg_i(p) = g_i'(p) \, dp$. We will show later that this deposit of numéraire is necessary and sufficient for the exchange to be budget-balanced or solvent, i.e., the exchange system does not extend credit.

The exchange mechanism maintains the demand curves of the LPs, along with the current price p_0. Assuming that n liquidity providers have contributed to the exchange their demand curves along with respective payments of risky asset and numéraire, the *aggregate demand curve* (i.e., the total quantity of risky asset that the exchange will hold at any given price) is given by the non-increasing function

$$g(p) = \sum_{i=1}^{n} g_i(p) \,. \tag{2}$$

Addition and removal of liquidity (LP "mints" and "burns", as they are known in practice) simply occur through additions and removals of particular g_i's to the aggregate demand curve of the exchange. These demand curves of the LPs can arise through *bonding curves* of traditional CFMMs (i.e., functions f such that the holdings of the joint pool (x, y) satisfy $f(x, y) = c$ for some c) but this is not necessary; i.e., the exchange mechanisms defined by our framework strictly generalize AMMs.

Trading A liquidity demanding trader who wants to trade with the exchange will do so by specifying a target (new) price $p_1 \neq p_0$. The trader gets a quantity $g(p_0) - g(p_1)$ of risky asset, and pays the following amount in numéraire:

$$-\int_{p_0}^{p_1} p \, dg(p) \,, \tag{3}$$

as determined by the aggregate liquidity of the exchange $g(p)$ of Eq. (2). As was the case for Eq. (1), this integral is well-defined, it is non-negative if $p_1 \geq p_0$, and non-positive if $p_1 \leq p_0$.

Uniswap v2 example To give a simple example, the particular case of a constant product market maker (CPMM), such as Uniswap v2, arises from our mechanism as follows: restrict the set of allowable demand curves g_i that an LP may submit to the form

$$g_i(p) = \frac{c_i}{\sqrt{p}} \,,$$

for some constant $c_i > 0$. Then, the aggregate demand curve of the exchange will be of the form

$$g(p) = \sum_{i=1}^{n} g_i(p) = \frac{c}{\sqrt{p}} \,,$$

for $c = \sum_{i=1}^{n} c_i > 0$. A trader who will trade with this exchange at a current price p_0 with a target price p_1 (or equivalently, with a specific quantity of risky asset to be purchased, since there a one-to-one correspondence) will obtain a quantity $g(p_0) - g(p_1) = c\left(\frac{1}{\sqrt{p_0}} - \frac{1}{\sqrt{p_1}}\right)$ of risky asset, and pay in numéraire

$$-\int_{p_0}^{p_1} p g'(p) \, dp = \int_{p_0}^{p_1} \frac{c}{2\sqrt{p}} \, dp = c\left(\sqrt{p_1} - \sqrt{p_0}\right) \,.$$

Comparing this to the same expressions for an "$xy = k$" CPMM, the trader gets exactly the same quantity of risky asset and pays exactly the same amount of numéraire as they would in the "$xy = k$" CPMM, with $k = c^2$. Essentially, the curve $g(p)$ above is just a reparameterization of the CPMM curve $xy = k$ in terms of prices [5] where the risky asset is available in quantity x in the pool and the amount of numéraire is y[4].

[4] In particular, $x = g(p) = c/\sqrt{p}$ and $y = c\sqrt{p}$ at all times in the pool for the corresponding defined price p.

Significance of LPs' demand curves In this mechanism, we view the individual demand curves chosen by the LPs as their *ideal preferences* with respect to risky asset holdings at each price in regards to their market making activity. They are in some sense "forced" to make the market —this is tautologically the reason that they participate in the exchange as LPs[5]— but *exactly how* they do this is specified by the shape of their demand curves. The requirement that each g_i be non-increasing can be explained through this argument: each demand curve of any LP has to always correspond to making the market; as the price of the risky asset increases, a market maker may only decrease their holdings of the asset (i.e., sell the asset), because if at any given price their holdings as defined in the exchange mechanism marginally increased (i.e., the LP would buy the risky asset at the marginal price), then any trader would sweep such a marginal quantity as it is to their advantage.

2.2 Price Discovery and Budget Balance

In the previous section, we defined a framework for an exchange mechanism. In order for an exchange to be reasonable, two properties would be necessary: (1) price discovery should occur, i.e., given an outside market with a fixed external market price, the exchange's price should eventually become identical to the market price; and (2) the exchange should at no point in time become insolvent, i.e., any feasible trade should always keep the amount of numéraire non-negative. (Because demand curves are non-negative, the amount of the risky asset is automatically non-negative.) Equivalently, the second property is broadly known in financial markets as a "no credit" requirement, i.e., that the exchange does not incorporate the ability of LPs to take credit. In the remainder of the section, we formalize and prove these properties for our model.

Proposition 1 (Price discovery). *If there exists an outside market with fixed external market price p of the risky asset with respect to the numéraire, then external market participants (arbitrageurs) always have financial incentive to trade with an exchange defined as per the framework of Section 2.1 until the price of such exchange becomes equal to the external market price.*

Proposition 2 (Budget balance). *An exchange defined as in the framework of Section 2.1 is budget-balanced or solvent, i.e., the amount of numéraire that the joint pool contains at all times (with any sequence of feasible trades, or liquidity additions/removals) is non-negative.*

We defer the full proofs of these two propositions to Appendix A.

[5] Note that LPs may also hold other portfolios of the risky asset, which of course need not be restricted to be non-increasing in the asset price, but their individual demand curves when they participating in an exchange mechanism need to reflect exactly and only the activity of making the market.

3 Exchange Description Complexity & Examples

Our general model in Section 2.1 allows LPs to submit arbitrary downward-sloping demand curves. Such curves are not generally representable in a finite amount of space, so practical considerations suggest restricting the space of demand curves that LPs are allowed to submit. We will say that an *exchange mechanism* is a restriction of the general exchange framework of Section 2.1 in which each LP demand curve is required to belong to a set of allowable demand curves, i.e., $g_i \in \mathcal{G}$ for some class $\mathcal{G}$ of non-increasing, non-negative functions over the positive reals. An exchange mechanism, then, is defined by the choice of class $\mathcal{G}$.

Towards defining a measure of exchange complexity, we will be interested in succinct ways of representing all the demand functions g in a class $\mathcal{G}$. Specifically, given an arbitrary such class $\mathcal{G}$, we can consider its conical hull. This is the smallest convex cone that contains[6] $\mathcal{G}$ or, equivalently, the closure of $\mathcal{G}$ under finite non-negative linear combinations:

$$\text{cone}(\mathcal{G}) = \left\{ \sum_{i=1}^{k} c_i g_i(p) : g_i(p) \in \mathcal{G}, c_i \geq 0, k \in \mathbb{N} \right\} .$$

In our context, non-negative linear combinations can be interpreted as aggregations of multiple LP positions.

A *basis* of a cone is a minimum-cardinality set of elements that generates the cone, meaning a set $\mathcal{S}$ such that $\text{cone}(\mathcal{S}) = \text{cone}(\mathcal{G})$. We then define the **exchange complexity** of an exchange (i.e., a choice $\mathcal{G}$ of allowable demand functions) as the cardinality of a basis for $\text{cone}(\mathcal{G})$.[7] By definition, if a set $\mathcal{G}$ of demand functions has exchange complexity k, every function of $\mathcal{G}$ can be represented by a k-tuple of non-negative real numbers (one coefficient for each of the basis functions).[8]

Our measure of exchange complexity is, by design, well defined for an arbitrary collection $\mathcal{G}$ of allowable demand functions. In all the real-world examples that we are aware of, this set $\mathcal{G}$ is already closed under non-negative linear combinations (i.e., is a cone). In this case, exchange complexity effectively counts an exchange's "primitive" LP positions from which all possible aggregations of LP positions can be derived.

[6] This definition makes sense because the intersection of convex cones is again a convex cone; see, e.g., Rockafellar [32] for further background.

[7] While our formalism in principle accommodates exchanges with infinite exchange complexity, any practical exchange needs to be defined by a finite basis on any compact (sub-)domain. Additionally, our results only make use of exchanges that have a finitely generated conic closure to approximate any demand curve within a finite approximation error under reasonable assumptions about the error metrics.

[8] The focus of this work is on information-theoretic complexity – approximation trade-offs, and we do not explicitly model computation. However, our positive results only make use of mechanisms for which computation with basis functions is straightforward.

This definition of exchange complexity allows us to formalize the intuition that some exchanges are easier to represent than others (e.g., that CFMMs are simpler than LOBs). Next, we evaluate the exchange complexity of all of the most popular types of exchanges used to trade crypto assets.

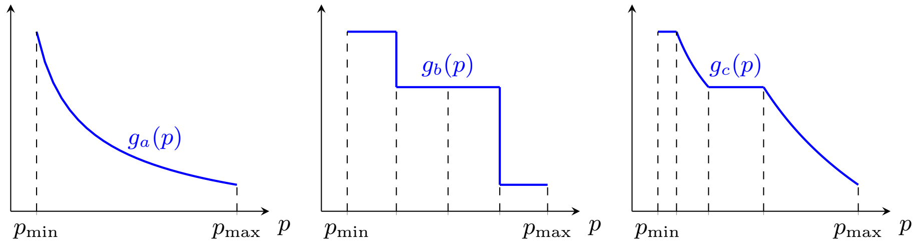

Fig. 1. $g \in \text{cone}(\mathcal{G})$ for three typical cases: (a) CPMM, (b) LOB, (c) Uniswap v3

CFMMs CFMMs are generated by the restriction to non-negative scalar multiples of a *single* basis function, i.e., $\mathcal{G} = \{c \cdot g(p) : c \geq 0\}$, where $g(p)$ is *one* reference demand curve, out of all the possible curves of the CFMM. The coefficient c of this basis function can then be interpreted as the liquidity parameter. As an example, for the CPMM, we can choose $g(p) = 1/\sqrt{p}$ (cf., Figure 1a); the coefficient can be interpreted as $\sqrt{k}$ for the k in "$xy = k$." In general, irrespective of the bonding curve, the exchange complexity of a CFMM is 1. Under standard assumptions (e.g., as in Angeris et al. [4]) on a CFMM's bonding curve f, the corresponding basis function g can be derived from f in a mechanical way, through optimization.

LOBs Limit order books consist of limit orders, which are (buy or sell) orders of quantities of the risky asset at some price. The predetermined prices at which limit orders can be specified are called *ticks*. In our framework, limit orders can be represented by a set of basis functions in which each function corresponds to a limit order at a specific tick (i.e., a step function, where the step occurs at the tick). According to our definition of exchange complexity above, then, the exchange complexity of a limit order book (cf., Figure 1b) with k ticks is k. If we restrict our attention to a price range $[p_{\min}, p_{\max}]$ with ticks $p_{\min}$, $p_{\min} + \epsilon$, $p_{\min} + 2\epsilon$, $\ldots$, $p_{\max}$, the exchange complexity of such a LOB would be $(p_{\max} - p_{\min})/\epsilon$.

There is a superficial difference in convention between traditional LOBs and our model of them in the preceding paragraph, concerning the *default action* after a trade that crosses the price of a limit order. In an LOB, the matching limit order would be automatically removed from the order book, whereas in our framework here the corresponding LP would, in effect, automatically place a new limit order in the opposite direction at the same price. In other words, a

LOB basis function is equivalent to both a limit buy and a limit sell at the tick price, and which one takes effect depends on the current price p_0 and the trade to be executed. Because limit orders can be easily added to or removed from traditional LOBs, and because our model accommodates LP mints and burns, there is no material difference between the two viewpoints.

Uniswap v3 Uniswap v3 (cf., Figure 1c) can be viewed as a hybrid of a CFMM and a LOB, with the CPMM curve applied only within a short price interval (in between two of the pre-defined ticks). By allowing multiple intervals, Uniswap v3 allows concentrated positions in the spirit of LOBs, a property known as *concentrated liquidity.* If there are k ticks contained in the interior of an interval $[p_{\min}, p_{\max}]$, then Uniswap v3's complexity on this interval is k. (There is one basis function for each price segment $[t_i, t_{i+1}]$ between two successive ticks; the function is constant up until the interval, decreases as in a CPMM within the interval, and is zero after the interval, as in Eq. (4)).

$$g_i(p) = \begin{cases} \frac{1}{\sqrt{t_i}} - \frac{1}{\sqrt{t_{i+1}}}\,, & \text{for } p \leq t_i \\ \frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_{i+1}}}\,, & \text{for } t_i \leq p \leq t_{i+1} \\ 0\,, & \text{for } p \geq t_{i+1} \end{cases} \tag{4}$$

Thus, the exchange complexity of both LOBs and Uniswap v3 is controlled by the number of ticks (independent of the spacing between them). In practice, ticks are sparser in Uniswap v3 than in a traditional LOB, and the former accordingly has lower exchange complexity than the latter. For an example calculation, if the ticks in Uniswap v3 are assumed to be of the form 1.0001^i, and $p_{\min} = 1.0001^s, p_{\max} = 1.0001^{s+t}$, then Uniswap v3's complexity in the price interval $[p_{\min}, p_{\max}]$ is

$$t = \frac{\log(p_{\max}/p_{\min})}{\log 1.0001} \approx 10000.5 \log(p_{\max}/p_{\min})\,.$$

We note that range orders in Uniswap v3 correspond to sums of single-interval positions (with one position per interval in the range) and are therefore automatically included in the cone generated by the basis functions defined above.

4 Complexity – Approximation Trade-offs

4.1 Notions of Approximation

Having defined the complexity of an exchange mechanism, we turn to defining the *expressiveness* of such a mechanism and proving fundamental trade-offs between complexity and expressiveness. Informally, we will measure the expressiveness of an exchange mechanism via the extent to which its allowable demand curves (i.e., the functions in the class $\mathcal{G}$) can represent arbitrary LP preferences (i.e., an arbitrary demand curve).

Precisely, denote by $\mathcal{F}$ the class of all non-increasing functions $f : [p_{\min}, p_{\max}] \to [f_{\min}, f_{\max}]$. This is the most general class of bounded demand curves according to our framework. Any arbitrary (bounded) preference of an LP will be some specific non-increasing function $f \in \mathcal{F}$.[9] We next define the extent to which some allowable demand curve $g \in \mathcal{G}$ (with the same domain and range) approximates f. (In this section we use g rather than g_i to denote an arbitrary function of $\mathcal{G}$.)[10]

First, we introduce the weighted ℓ_p norm in the function space as a distance metric; without loss of generality, assume we have a normalized (and integrable) weight function $w : [p_{\min}, p_{\max}] \to \mathbb{R}^+$ such that $\int_{p_{\min}}^{p_{\max}} w(p)\, dp = 1$. Then, the weighted ℓ_p distance of two functions $f, g \in \mathcal{F}$ is

$$d(f,g) = \left(\int_{p_{\min}}^{p_{\max}} w(s) \left| f(s) - g(s) \right|^p \, ds \right)^{1/p} .$$

The weight function w can be interpreted as a measure on the price space, for example reflecting a belief (by an LP, the AMM designer, or the community) that some prices may be more relevant than others. On a first read, we encourage the reader to take w to be the constant function $w(s) = 1/(p_{\max} - p_{\min})$ for all $s \in [p_{\min}, p_{\max}]$.

Given this definition, we define the **approximation error** of the exchange defined by $\mathcal{G}$ as the worst-case (over arbitrary LP preferences/demand curves $f \in \mathcal{F}$) distance from the best-case approximation (over allowable functions $g \in \text{cone}(\mathcal{G})$) of f, as above:

$$\text{err}(\mathcal{G}) = \sup_{f \in \mathcal{F}} \left\{ \inf_{g \in \text{cone}(\mathcal{G})} d(f,g) \right\} . \tag{5}$$

4.2 Upper and Lower Bounds

From the AMM designer's perspective, an "optimal" AMM would enable LPs to have their preferences expressed closely; a bit more formally, the worst-case

[9] Note that in what follows f is a demand curve, as defined in Section 2.1, and not a bonding curve of a CFMM.

[10] The restricting to a bounded domain and range is convenient but can be relaxed considerably. The fundamental issue is that, to meaningfully speak about function approximations and avoid infinite distances between distinct functions, we need to impose constraints on allowable demand functions and/or the choice of distance function and underlying measure (on prices). Functions with bounded domain and range are convenient because they are integrable no matter what the distance notion and measure. Our results can be generalized by considering combinations of demand function classes and classes of measures for which the same integrability properties are guaranteed.

Additionally, it will be apparent from our lower bound (Theorem 2) that, if the family of functions $\mathcal{F}$ was not bounded by some finite bound $f_{\max} < \infty$, there would be no finite approximation error guarantee with any finite complexity (under any natural notion of approximation error).

approximation error through the AMM for arbitrary LP demand curves should be low, and intuitively should decrease with the complexity of the exchange mechanism: the higher exchange complexity should result in a payoff of lower worst-case approximation error. The results below characterize this trade-off, by identifying the best-possible worst-case approximation error as a function of the exchange complexity. For example, for the special case in which the approximation metric between two functions is the (unweighted) ℓ_1 distance, an exchange complexity (equivalently, number of basis functions) of $\Theta(1/\epsilon)$ is necessary and sufficient to achieve an ϵ worst-case approximation error.

Our upper bound argument also implies the (intuitive but previously unformalized) fact that limit order books at appropriately defined price ticks attain the optimal approximation error guarantee for a given level of exchange complexity (up to a factor of 2). In other words, when computation and storage are not first-order constraints, LOBs are nearly optimally expressive exchange mechanisms.

Theorem 1 (Upper bound). *For every $\epsilon > 0$, there exists a limit order book (LOB) exchange mechanism $\mathcal{G}$ with exchange complexity $k = \mathrm{O}(1/\epsilon^p)$ that attains approximation error*

$$err(\mathcal{G}) \leq \epsilon \cdot \frac{f_{max} - f_{min}}{2} .$$

Theorem 2 (Lower bound). *For every $\epsilon > 0$, every exchange mechanism $\mathcal{G}$ with exchange complexity $o(1/\epsilon^p)$ suffers approximation error*

$$err(\mathcal{G}) \geq \epsilon \cdot \frac{f_{max} - f_{min}}{4} .$$

For the detailed proofs of Theorems 1 and 2 we refer to Sections 6.1 and 6.2 respectively.

5 Uniswap v3

Next, we answer the question: to what extent do various formats in practice come close to this complexity – approximation trade-off? Historically, constant product market makers (CPMMs) were first built for gas efficiency purposes [1], but when it was realized that this came often at the expense of capital efficiency, the proposal of Uniswap v3 came around [2], which trades like a CPMM curve inside tight intervals at a pre-defined tick spacing, which are otherwise independent. In this section, we consider Uniswap v3, which is at the time of writing a widely used AMM, as an enlightening example to showcase how our theory can be applied to formally prove approximation guarantees for AMMs employed in practice.

In particular, we are able to prove for the first time that —under a particular assumption of the returns distribution with maximum entropy, i.e., a uniform prior in the returns space— a variation of Uniswap v3 with variable tick spacing δ obtains the optimal[11] approximation error for arbitrary LP preferences (demand

[11] up to a constant multiplicative factor of 4

curves), or equivalently, can approximate arbitrarily closely any other CFMM curve. The precise formulation follows.

Theorem 3. *For every $\epsilon > 0$, there exists a Uniswap v3-like exchange mechanism $\mathcal{G}$ with $n = \mathrm{O}(1/\epsilon^p)$ ticks at prices $p_{min}(1+\delta)^i$ for $i \in \{0, 1, \ldots, n\}$ where $\log(1+\delta) = \epsilon^p \log(p_{max}/p_{min})$, that attains approximation error according to Eq.* (5) *with a normalized weight function $w(p)$ which assigns measure at most $\mathrm{O}(1/n)$ to each of the intervals defined by these ticks, of*

$$err(\mathcal{G}) \leq \mathrm{O}(\epsilon \cdot (f_{max} - f_{min}))\,.$$

The detailed proof of Theorem 3 is relegated to Section 6.3.

6 Proofs

6.1 Proof of Theorem 1

Let $\epsilon > 0$, and a normalized weight function $w\colon [p_{\min}, p_{\max}] \to \mathbb{R}^+$ such that $\int_{p_{\min}}^{p_{\max}} w(p)\, dp = 1$. Then, since $w(p) \geq 0$ $\forall p \in [p_{\min}, p_{\max}]$, split the interval $[p_{\min}, p_{\max}]$ into $n = 1/\epsilon^p$ equal measure (according to the weight function) sub-intervals $[t_i, t_{i+1}]$, $\forall i \in \{1, 2, \ldots, n\}$, i.e., such that $\int_{t_i}^{t_{i+1}} w(p)dp = \frac{1}{n}$. Define the limit order book (LOB) exchange mechanism $\mathcal{G} = \mathrm{cone}(\mathcal{G})$ as the conical hull of the following set of basis functions: each basis function represents a limit order at each price point t_i above, i.e., the basis function is a unit step function dropping from 1 to 0 at price t_i. The exchange complexity of this $\mathcal{G}$ is therefore $1/\epsilon^p$.

Consider any $f \in \mathcal{F}$, and define the following $g_f \in \mathrm{cone}(\mathcal{G})$ that will "approximate" this f:

$$\forall p \in (t_i, t_{i+1}),\ g_f(p) = \frac{f(t_i) + f(t_{i+1})}{2}\,. \tag{6}$$

It is true that this $g_f \in \mathrm{cone}(\mathcal{G})$, because g_f is piecewise constant, with function value drops occurring only at the prices t_i (see Figure 1b for an example representation).

We have that

$$\forall p \in (t_i, t_{i+1}),\ |f(p) - g_f(p)| \leq \frac{f(t_i) - f(t_{i+1})}{2}\,,$$

since f is non-increasing, and by the definition of g_f in Eq. (6).

Hence, we obtain the desired result:

$$\begin{aligned}
\mathrm{err}(\mathcal{G}) = \sup_{f\in\mathcal{F}} \left\{ \inf_{g\in\mathrm{cone}(\mathcal{G})} d(f,g) \right\} &\le \sup_{f\in\mathcal{F}} \left(\sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} w(s) \left| f(s) - g_f(s) \right|^p ds \right)^{1/p} \\
&\le \sup_{f\in\mathcal{F}} \left(\sum_{i=1}^{n} \int_{t_i}^{t_{i+1}} w(s) \left(\frac{f(t_i) - f(t_{i+1})}{2} \right)^p ds \right)^{1/p} \\
&= \frac{1}{2n^{1/p}} \sup_{f\in\mathcal{F}} \left(\sum_{i=1}^{n} [f(t_i) - f(t_{i+1})]^p \right)^{1/p} \\
&\le \frac{1}{2n^{1/p}} \sup_{f\in\mathcal{F}} \sum_{i=1}^{n} [f(t_i) - f(t_{i+1})] \\
&\le \epsilon \cdot \frac{f_{\max} - f_{\min}}{2} ,
\end{aligned}$$

where the second-to-last inequality follows from the inequality between ℓ_1 and ℓ_p norms in the function space.

6.2 Proof of Theorem 2

Let $\epsilon > 0$, and a normalized weight function $w\colon [p_{\min}, p_{\max}] \to \mathbb{R}^+$ such that $\int_{p_{\min}}^{p_{\max}} w(p)\, dp = 1$. Similarly to the upper bound, but with double the amount of intervals, split the interval $[p_{\min}, p_{\max}]$ into $2n$ (where $n = 1/\epsilon^p$) equal measure (according to the weight function) sub-intervals $[t_i, t_{i+1}]$, $\forall i \in \{1, 2, \ldots, 2n\}$, i.e., such that $\int_{t_i}^{t_{i+1}} w(p) dp = \frac{1}{2n}$. Now, consider any exchange mechanism $\mathcal{G}$ with exchange complexity $\le \frac{1}{\epsilon^p} - 1$, i.e., such that cone($\mathcal{G}$) is generated by $\le \frac{1}{\epsilon^p} - 1$ basis functions; suppose without loss of generality that these are $g_1, g_2, \ldots, g_{n-1} \in \mathrm{cone}(\mathcal{G})$.

Lemma 1. *For every basis function g_i (where $i \in \{1, 2, \ldots, n-1\}$ as above), there exists at most one interval of the form $[t_{2l+1}, t_{2l+3}]$ for some l (where t's are defined as in the above paragraph) such that*

$$g_i(t_{2l+1}) - g_i(t_{2l+3}) > \frac{g_i(p_{min}) - g_i(p_{max})}{2} .$$

Proof. Let g_i be any basis function. Assume that the lemma's hypothesis is not true, i.e., there exist at least two intervals $[t_{2l+1}, t_{2l+3}]$ and $[t_{2m+1}, t_{2m+3}]$ for some l, m such that the lemma's equation holds for each of these intervals. But since g_i is non-increasing, this would necessitate that

$$\begin{aligned}
g_i(p_{\min}) - g_i(p_{\max}) &\ge \big[g_i(t_{2l+1}) - g_i(t_{2l+3})\big] + \big[g_i(t_{2m+1}) - g_i(t_{2m+3})\big] \\
&> g_i(p_{\min}) - g_i(p_{\max}) ,
\end{aligned}$$

which completes the proof by contradiction.

From Lemma 1 and the pigeonhole principle (there exist n odd-indexed intervals of the form $[t_{2l+1}, t_{2l+3}]$ for some l, but only $n-1$ basis functions), we get that there exists at least one interval $[t_{2l+1}, t_{2l+3}]$ (for some l) such that for all $i \in \{1, 2, \dots, n-1\}$,

$$g_i(t_{2l+1}) - g_i(t_{2l+3}) \le \frac{g_i(p_{\min}) - g_i(p_{\max})}{2} \,,$$

and because $\operatorname{cone}(\mathcal{G})$ is finitely generated, it holds that for all $g \in \operatorname{cone}(\mathcal{G})$,

$$g(t_{2l+1}) - g(t_{2l+3}) \le \frac{g(p_{\min}) - g(p_{\max})}{2} \,. \tag{7}$$

Consider the following specific $f_a \in \mathcal{F}$:

$$f_a(p) = \begin{cases} f_{\max} \,, & \text{for } p_{\min} \le p < t_{2l+2} \\ f_{\min} \,, & \text{for } t_{2l+2} \le p \le p_{\max} \end{cases} \,.$$

Then, for all $g \in \operatorname{cone}(\mathcal{G})$, we distinguish 3 cases:[12]

- If $g(t_{2l+1}) \ge f_{\max}$, then $g(t_{2l+2}) \ge g(t_{2l+3}) \ge \frac{f_{\max}+f_{\min}}{2}$, and by Eq. (7),
$$\int_{t_{2l+2}}^{t_{2l+3}} w(s)\,|f_a(s) - g(s)|^p\,ds \ge \frac{(f_{\max} - f_{\min})^p}{n \cdot 2^{1+p}} \,.$$
- If $g(t_{2l+3}) \le f_{\min}$, then $g(t_{2l+2}) \le g(t_{2l+1}) \le \frac{f_{\max}+f_{\min}}{2}$, and by Eq. (7),
$$\int_{t_{2l+1}}^{t_{2l+2}} w(s)\,|f_a(s) - g(s)|^p\,ds \ge \frac{(f_{\max} - f_{\min})^p}{n \cdot 2^{1+p}} \,.$$
- Otherwise, for some $\delta_1, \delta_2 > 0$ we have that $f_{\min} < f_{\min} + \delta_2 = g(t_{2l+3}) \le g(t_{2l+1}) = f_{\max} - \delta_1 < f_{\max}$; then Eq. (7) becomes $\delta_1 + \delta_2 \ge \frac{f_{\max}-f_{\min}}{2}$, and we have that
$$\int_{t_{2l+1}}^{t_{2l+3}} w(s)\,|f_a(s) - g(s)|^p\,ds \ge \frac{\delta_1^p + \delta_2^p}{2n} \ge \frac{(\delta_1 + \delta_2)^p}{n \cdot 2^p} \ge \frac{(f_{\max} - f_{\min})^p}{n \cdot 4^p} \,,$$
where the second-to-last inequality follows from Hölder's inequality.

Hence, we obtain the desired result:

$$\begin{aligned} \operatorname{err}(\mathcal{G}) = \sup_{f \in \mathcal{F}} \left\{ \inf_{g \in \operatorname{cone}(\mathcal{G})} d(f,g) \right\} &\ge \inf_{g \in \operatorname{cone}(\mathcal{G})} \left(\int_{p_{\min}}^{p_{\max}} w(s)\,|f_a(s) - g(s)|^p\,ds \right)^{1/p} \\ &\ge \epsilon \cdot \frac{f_{\max} - f_{\min}}{4} \,. \end{aligned}$$

[12] Note that the first two cases exist because $g \in \operatorname{cone}(\mathcal{G})$ is not restricted to $[f_{\min}, f_{\max}]$; it is potentially allowed to be led outside in order to approximate f_a optimally.

6.3 Proof of Theorem 3

Let $\epsilon > 0$, and consider ticks $t_i = p_{\min}(1+\delta)^i$ for $i \in \{0, 1, \dots, n\}$ where $\log(1+\delta) = \epsilon^p \log(p_{\max}/p_{\min})$, and $n = \log(p_{\max}/p_{\min})/\log(1+\delta)$, so that $t_0 = p_{\min}$ and $t_n = p_{\max}$. Consider the normalized weight function $w\colon [p_{\min}, p_{\max}] \to \mathbb{R}^+$ such that $\int_{p_{\min}}^{p_{\max}} w(p)\, dp = 1$, with the property that for some constant $C > 0$, $\forall i \in \{0, 1, \dots, n-1\}$, $\int_{t_i}^{t_{i+1}} w(p)\, dp \leq \frac{C^p}{n}$. Our Uniswap v3-like exchange mechanism $\mathcal{G} = \text{cone}(\mathcal{G})$ is described with the following $n+1$ basis functions: one basis function for each of the intervals $[t_i, t_{i+1}]$ for $i \in \{0, 1, \dots, n-1\}$ defined by

$$g_i(p) = \begin{cases} \frac{1}{\sqrt{t_i}} - \frac{1}{\sqrt{t_{i+1}}}\,, & \text{for } p_{\min} \leq p \leq t_i \\ \frac{1}{\sqrt{p}} - \frac{1}{\sqrt{t_{i+1}}}\,, & \text{for } t_i \leq p \leq t_{i+1} \\ 0\,, & \text{for } t_{i+1} \leq p \leq p_{\max} \end{cases} \,,$$

along with the additional basis function $g_n(p)$ that is everywhere 1^{13}.

Consider any $f \in \mathcal{F}$, and define the following $g_f \in \text{cone}(\mathcal{G})$ that will "approximate" this f:

$$g_f(p) = f(p_{\max}) g_n(p) + \sum_{i=0}^{n-1} \frac{f(t_i) - f(t_{i+1})}{\frac{1}{\sqrt{t_i}} - \frac{1}{\sqrt{t_{i+1}}}} g_i(p)\,.$$

Then, it holds that

$$\forall p \in (t_i, t_{i+1}),\ |f(p) - g_f(p)| \leq f(t_i) - f(t_{i+1})\,.$$

Hence, we obtain the stated result by a similar argument to that of Section 6.1.

Acknowledgements We thank anonymous reviewers of the Financial Cryptography and Data Security conference for useful suggestions. The first author is supported in part by NSF awards CNS-2212745, CCF-2212233, DMS-2134059, and CCF-1763970. The second author is supported by the Briger Family Digital Finance Lab at Columbia Business School. The third author's research at Columbia University is supported in part by NSF awards CNS-2212745, and CCF-2006737.

Disclosures The first author is a Research Fellow with automated market making protocols, including ones mentioned in this work. The second author is an advisor to fintech companies. The third author is Head of Research at a16z Crypto, a venture capital firm with investments in automated market making protocols.

[13] Note that this additional basis function is always necessary whenever $f_{\min} \neq 0$ to obtain an *arbitrarily good* approximation of any curve, due to the construction of the Uniswap curves to end at exactly 0 at the end of each interval.

References

1. Adams, H., Zinsmeister, N., Robinson, D.: Uniswap v2 core (2020)
2. Adams, H., Zinsmeister, N., Salem, M., Keefer, R., Robinson, D.: Uniswap v3 core (2021)
3. Angeris, G., Chitra, T.: Improved price oracles: Constant function market makers. In: Proceedings of the 2nd ACM Conference on Advances in Financial Technologies. pp. 80–91 (2020)
4. Angeris, G., Evans, A., Chitra, T.: Replicating market makers. arXiv preprint arXiv:2103.14769 (2021)
5. Angeris, G., Evans, A., Chitra, T.: Replicating monotonic payoffs without oracles. arXiv preprint arXiv:2111.13740 (2021)
6. Angeris, G., Evans, A., Chitra, T., Boyd, S.: Optimal routing for constant function market makers. In: Proceedings of the 23rd ACM Conference on Economics and Computation. pp. 115–128 (2022)
7. Angeris, G., Kao, H.T., Chiang, R., Noyes, C., Chitra, T.: An analysis of uniswap markets. arXiv preprint arXiv:1911.03380 (2019)
8. Aoyagi, J.: Liquidity provision by automated market makers. SSRN 3674178 (2020)
9. Aoyagi, J., Ito, Y.: Coexisting exchange platforms: Limit order books and automated market makers. SSRN 3808755 (2021)
10. Barbon, A., Ranaldo, A.: On the quality of cryptocurrency markets: Centralized versus decentralized exchanges. arXiv preprint arXiv:2112.07386 (2021)
11. Bichuch, M., Feinstein, Z.: Axioms for automated market makers: A mathematical framework in fintech and decentralized finance (2022), https://arxiv.org/abs/2210.01227
12. Buterin, V.: Let's run on-chain decentralized exchanges the way we run prediction markets (Oct 2016), www.reddit.com/r/ethereum/comments/55m04x/lets_run_onchain_decentralized_exchanges_the_way/
13. Capponi, A., Jia, R.: The adoption of blockchain-based decentralized exchanges. arXiv preprint arXiv:2103.08842 (2021)
14. Chitra, T., Angeris, G., Evans, A.: How liveness separates cfmms and order books (2021)
15. Ciampi, M., Ishaq, M., Magdon-Ismail, M., Ostrovsky, R., Zikas, V.: Fairmm: A fast and frontrunning-resistant crypto market-maker. In: International Symposium on Cyber Security, Cryptology, and Machine Learning. pp. 428–446. Springer (2022)
16. Engel, D., Herlihy, M.: Composing networks of automated market makers. In: Proceedings of the 3rd ACM Conference on Advances in Financial Technologies. pp. 15–28 (2021)
17. Engel, D., Herlihy, M.: Presentation and publication: Loss and slippage in networks of automated market makers. arXiv preprint arXiv:2110.09872 (2021)
18. Fan, Z., Marmolejo-Cossío, F., Altschuler, B., Sun, H., Wang, X., Parkes, D.C.: Differential liquidity provision in uniswap v3 and implications for contract design. arXiv preprint arXiv:2204.00464 (2022)

19. Felekis, G., Kristensen, J.: λ - constant function markets generalizing and mixing automated market makers. In: 2022 IEEE International Conference on Blockchain (Blockchain). pp. 290–297 (2022)
20. Forgy, E., Lau, L.: A family of multi-asset automated market makers. arXiv preprint arXiv:2111.08115 (2021)
21. Goyal, M., Ramseyer, G., Goel, A., Mazières, D.: Batch exchanges with constant function market makers: Axioms, equilibria, and computation. arXiv preprint arXiv:2210.04929 (2022)
22. Huynh, Y.: Providing liquidity in uniswap v3 (2022)
23. Jensen, J.R., Pourpouneh, M., Nielsen, K., Ross, O.: The homogenous properties of automated market makers. arXiv preprint arXiv:2105.02782 (2021)
24. Kaiko: Crypto Markets Recover Despite 9.1% Inflation (Jul 2022), https://blog.kaiko.com/crypto-markets-recover-despite-9-1-inflation-9d7db87ab83f
25. Krishnamachari, B., Feng, Q., Grippo, E.: Dynamic automated market makers for decentralized cryptocurrency exchange. In: 2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–2 (2021)
26. Lehar, A., Parlour, C.A.: Decentralized exchanges. Tech. rep., Working paper (2021)
27. Lu, A., Köppelmann, M.: Building a Decentralized Exchange in Ethereum (Mar 2017), https://blog.gnosis.pm/building-a-decentralized-exchange-in-ethereum-eea4e7452d6e
28. Moosavi, M., Clark, J.: Lissy: Experimenting with on-chain order books. arXiv preprint arXiv:2101.06291 (2021)
29. Neuder, M., Rao, R., Moroz, D.J., Parkes, D.C.: Strategic liquidity provision in uniswap v3. arXiv preprint arXiv:2106.12033 (2021)
30. O'Hara, M.: Market microstructure theory. Blackwell, Malden, Mass., repr. edn. (2011)
31. Port, A., Tiruviluamala, N.: Mixing constant sum and constant product market makers. arXiv preprint arXiv:2203.12123 (2022)
32. Rockafellar, R.T.: Convex analysis. Princeton Landmarks in Mathematics and Physics, Princeton University Press, Princeton, NJ (Dec 1996)
33. Shuttleworth, D.: Serum: A Decentralized On-Chain Central Limit Order Book | ConsenSys Cryptoeconomic Research (2022), https://consensys.net/blog/cryptoeconomic-research/serum-a-decentralized-on-chain-central-limit-order-book/
34. Wang, S., Krishnamachari, B.: Optimal trading on a dynamic curve automated market maker. In: 2022 IEEE International Conference on Blockchain and Cryptocurrency (ICBC). pp. 1–5 (2022)
35. Wu, M., McTighe, W.: Constant power root market makers. arXiv preprint arXiv:2205.07452 (2022)
36. Xu, J., Paruch, K., Cousaert, S., Feng, Y.: Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols. arXiv preprint arXiv:2103.12732 (2021)
37. Yin, J., Ren, M.: On liquidity mining for uniswap v3. arXiv preprint arXiv:2108.05800 (2021)
38. Young, J.E.: On equivalence of automated market maker and limit order book systems (2020)

A Deferred Proofs of Section 2.2

Proof (Price discovery). Assume that the current price of the exchange is $p_0 \neq p$. Suppose that an external market participant comes to the exchange and is willing to trade to some price p_1, and then uses the external market to trade back. We prove that the maximum profits will be obtained at $p_1 = p$; therefore, if the trader does not maximize their profits, other external market participants will continue to have an incentive to trade until the price of the exchange is p and the conclusion follows.

Due to Eq. (3), the external market participant's optimization problem for their profit is:

$$\max_{p_1 \in \mathbb{R}^+} p(g(p_0) - g(p_1)) + \int_{p_0}^{p_1} p dg(p) = \max_{p_1 \in \mathbb{R}^+} (p_1 - p)g(p_1) - \int_0^{p_1} g(p)dp$$

First-order conditions then prove that the optimum is attained at $p_1 = p$.

Proof (Budget balance). Assume that the current price of the exchange is p_0. First, we note that liquidity additions and removals, due to the linear nature of the aggregate demand curves and the numéraire contributed/removed by Eq. (1) with respect to the curves $g_i(p)$, do not affect the rest of the joint pool, i.e., if the amount of numéraire was non-negative before the operation, so it is after it. Trading is the only action which is yet unclear how it affects the amount of numéraire in the pool. In aggregate, the joint pool contains a quantity $g(p_0)$ of risky asset, and in numéraire by Eq. (1):

$$\sum_{i=1}^{n} - \int_0^{p_0} p dg_i(p) = - \int_0^{p_0} p dg(p) \geq 0 \,,$$

because g is non-increasing (as the sum of non-increasing functions) and $p_0 \geq 0$. Suppose that a trader comes and moves the pool price to p_1. The new amount of numéraire contained in the pool by the above equation and Eq. (3) is

$$- \int_0^{p_0} p dg(p) - \int_{p_0}^{p_1} p dg(p) = - \int_0^{p_1} p dg(p) \geq 0 \,,$$

thereby completing our argument.