Dynamic Curves for Decentralized Autonomous Cryptocurrency Exchanges

Bhaskar Krishnamachari ✉
Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

Qi Feng ✉
Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

Eugenio Grippo ✉
Viterbi School of Engineering, University of Southern California, Los Angeles, CA, USA

──── **Abstract** ────

One of the exciting recent developments in decentralized finance (DeFi) has been the development of decentralized cryptocurrency exchanges that can autonomously handle conversion between different cryptocurrencies. Decentralized exchange protocols such as Uniswap, Curve and other types of Automated Market Makers (AMMs) maintain a liquidity pool (LP) of two or more assets constrained to maintain at all times a mathematical relation to each other, defined by a given function or curve. Examples of such functions are the constant-sum and constant-product AMMs. Existing systems however suffer from several challenges. They require external arbitrageurs to restore the price of tokens in the pool to match the market price. Such activities can potentially drain resources from the liquidity pool. In particular dramatic market price changes can result in low liquidity with respect to one or more of the assets and reduce the total value of the LP. We propose in this work a new approach to constructing the AMM by proposing the idea of *dynamic curves*. It utilizes input from a market price oracle to modify the mathematical relationship between the assets so that the pool price continuously and automatically adjusts to be identical to the market price. This approach eliminates arbitrage opportunities and, as we show through simulations, maintains liquidity in the LP for all assets and the total value of the LP over a wide range of market prices.

1 Introduction

Since the introduction of Bitcoin as the first peer-to-peer digital cash [15], the birth of different cryptocurrencies has revolutionized the world of finance [21]. As of the time of writing this article, it is estimated that the total cryptocurrency market capitalization is more than $600 Billion, involving thousands of different coins [10].

Traditionally, cryptocurrency exchanges, which use an order book mechanism, are centralized. They suffer from concerns about the concentration of financial power [14] and being prone to a single point of failure, resulting in a potentially significant loss of funds when attacked [3]. Additionally, they also pose a liquidity problem for tokens with a smaller market capitalization resulting in barriers to entry to the financial market [14].

On the other hand, it is difficult to implement the order book model in a decentralized manner in the form of a blockchain smart contract [20] [3]. First, market makers will face high gas costs to execute transactions, regardless of their sizes [20]. Second, it will require a complex matching algorithm to support a variety of order types [3].

Automated Market Makers such as Hanson's logarithmic market scoring rules (LMSRs) are widely used in traditional prediction markets to address the problem of low liquidity and trading volume [13, 19]. An LMSR-based AMM is also used in decentralized prediction markets such as Gnosis [12] and Augur [16]. Given certain similar market characteristics, curve-based Automated Market Makers (AMM) were recently introduced to address the challenges in a currency exchange context. They are currently one of the areas of decentralized finance receiving the most attention. Instead of relying on the traditional market makers to provide liquidity, decentralized exchanges utilizing curve-based AMMs, such as Bancor [14], Uniswap [1], StableSwap/Curve [11] and many others implement a liquidity pool (LP) using smart contracts on a blockchain. In this model, liquidity providers supply single or multiple types of tokens to the designated liquidity pools, and traders exchange against the pools of tokens instead of relying on order matching. The liquidity pool of these AMMs track a pre-defined mathematical function (curve), thus determining how many tokens of one type to provide to a trader in exchange for a certain amount of another. Curve-based AMMs provide a continuous supply of liquidity compared to the order book model. Additionally, depending on the mathematical function (curve) utilized, they can potentially allow for a wide range of exchange prices. However, the token price within a liquidity pool for a given AMM (which we refer to as the pool price) might be different from the market price.

When such a gap occurs on a decentralized AMM-based exchange, arbitrageurs may have the opportunity to buy or sell tokens to set the pool price equal to the market price, restoring equilibrium. However, in some cases, particularly when the market price changes dramatically, the AMM-based LP could lose liquidity with respect to one or more of the assets. We propose in this work a new approach to constructing the AMM by proposing the idea of *dynamic curves*. It utilizes input from a market price oracle to modify the mathematical relationship between the assets so that the pool price continuously and automatically adjusts to be identical to the market price. This eliminates arbitrage opportunities and, as we show through simulations, helps the AMM-based LP maintain liquidity and total value over a wide range of market prices.

The following are the key contributions of this work:

- We present a simple and unified mathematical and conceptual framework (in section 3) describing existing curve-based AMMs and key metrics such as pool price, slippage, divergence loss. It unifies much of what is known about them today. We believe this section will be of independent interest to researchers starting out in this area.
- We focus on the liquidity problem posed by arbitrageurs on existing AMMs, especially when the market price for one of the assets becomes too high, causing asset depletion and value reduction of the liquidity pool.
- We present a new dynamic curve mechanism, which is general enough to be adapted to any monotonic function/curve used on an AMM. This mechanism relies on an external market price oracle and eliminates arbitrageurs. We illustrate the mechanism concretely through generalizations of both constant-sum and constant-product models.
- We present numerical simulations showing the clear advantages of our proposed dynamic curve mechanism in a) retaining greater liquidity in the pool to benefit small traders, b) retaining greater total value in the liquidity pool, and c) functioning effectively over a much larger range of market prices.

The rest of this paper is organized as follows: we present and discuss relevant prior work in section 2. In section 3 we give a unified treatment and definition of key concepts and metrics relevant to decentralized AMMs. In section 4 we propose and describe our new dynamic AMM models. In section 5 we present agent-based simulations and compare the

performance of four different AMMs, including two static AMMs (Constant- Sum AMM, Constant-Product AMM) and their two dynamic generalizations that we introduce in this work. We present concluding comments in section 6.

2 Related Work

Bancor was the first DEX to implement the type of AMM called Bonding Curve, which provides continuous liquidity [14]. In this type of AMM, there is a single token (Bancor Network Token - BNT) used as an intermediate currency. There are separate pools for each non-native currency to be traded against BNT. This model is a little different from the arbitrary two-asset curve based AMM's that we focus on in this paper (though there are significant connections as well). In curve-based AMMs, any two currencies could be traded directly against each other.

Borrowing solutions from the prediction market, Buterin [6] first proposed such a curve-based AMM for a decentralized exchange. Specifically, he proposed the Constant Product Curve. It is a convex curve that takes the form of $x \cdot y = k$, where x and y are the total supply of two tokens in a liquidity pool and k is the product constant. It was subsequently implemented by Adams *et al.* [1] to create Uniswap.

With the shape of a downward-sloping straight line, the Constant Sum Curve [5][7] takes the form of $x + y = k$. x and y are the total supply of two tokens in a liquidity pool, and k is the sum constant. StableSwap/Curve [11] implemented an AMM curve that is a blend of Constant Sum and Constant Product to provide continuous liquidity, price stability and a built-in pool balancing mechanism.

Wang [19] proposed the Constant Ellipse Curve AMM with the general form of $(x - a)^2 + (y - a)^2 + b \cdot xy = C$, in which a and b are constant. One can choose between the concave and the convex curve in the first quadrant [19]. Wang also presents the curve corresponding to the LMSR rule.

Angeris and Chitra analyze such curve-based AMMs, which they refer to as constant function market makers in the general case, i.e., with arbitrarily many tokens [2]. They analyze various mathematical properties of such AMMs, including formulating the optimal arbitrage by traders as a convex optimization problem.

2.1 Performance metrics for AMMs

Slippage and divergence loss are the two main factors contributing to the proposal and adoption of different AMMs. The former is directly tied to the loss of traders, while the latter is directly connected to the liquidity providers' returns.

Slippage is the difference between the expected and actual trade execution price [18], and in the AMM context, it is defined as the gap between the pool price before a trade and the effective price obtained for the trade (see section 3.3). As long as token price changes during trade, slippage incurs. In addition, when large trades happen compared to pool size, slippage increases dramatically, resulting in lower trading profits [9].

Divergence loss, sometimes called impermanent loss, incurs when liquidity providers withdraw liquidity with the presence of a difference in token price before and after trades [17]. If funds are pulled out during a large price swing, liquidity providers will suffer a loss of total asset value, compared to simply holding the assets [8]. Given trades might affect token pool prices and hence divergence loss, it is important to distinguish between regular and arbitrage trading. Divergence loss due to arbitrage trading in closing pool and market price gap can be mitigated by incorporating reliable oracles to protect liquidity providers [4].

2.2   Slippage and Divergence Loss in AMMs

In Bancor, given the dynamic pool price by design, trades experience slippage. Divergence loss incurs as the pool price is intrinsic and relies on arbitrageurs to close price gaps [14]. To protect liquidity providers, Bancor v2 integrates with Chainlink price oracle to reduce divergence loss from arbitraging [4].

In Uniswap, similar to Bancor, the pool price is inherently unstable and the size of trades in relation to pool size affects pool price to different extents. The larger the trades are, the higher slippage and divergence loss can occur.

The constant-sum curve has zero slippage [5, 7] and no divergence loss (as we show in section 3). However, because it has a fixed price and finite liquidity, it is only suitable for stablecoins and could easily be depleted of one of its pool assets; for this reason, it is primarily of theoretical interest [5, 7]. We use it as a baseline in our work.

StableSwap/Curve introduces an invariant that allows trading on a Constant Sum shaped curve when the portfolio is relatively balanced and switch trading to a Constant Product shaped curve when imbalanced [11]. Such a design allows much lower slippage and divergence loss but is only applicable to stablecoins as the price of the desired trading range is always close to 1.

The constant ellipse curve introduced by Wang [19] has a fixed price range compared to that of a Constant Product Curve and thus a fixed range of slippage and divergence loss. Wang also concludes that the LMSR curve would not be suitable for exchanges if the numbers of the two tokens are not balanced in the liquidity pool.

The proposal in this paper presents an approach for AMM-based decentralized exchanges using dynamic curves that eliminates the possibility of arbitrage and thus any divergence loss. Instead, as we show, depending on the chosen family of curves, any slippage loss incurred by traders is converted to an equivalent gain for the liquidity providers. A dynamic version of the constant sum curve is a special case of our proposed solution, and in that case, there is no slippage loss at all.

2.3   Simulation

There are two simulations conducted respectively on StableSwap/Curve and Uniswap v1 to evaluate the DEXes performance. Egorov [11] suggests that StableSwap/Curve generates 312% APR and 0.06% fee per trade for liquidity providers with total liquidity of $30000 in DAI, USDC and USTD over 6 months. Angeris *et al.* [3] conduct an agent-based simulation to test the hypothesis that Uniswap has a robust market mechanism with little arbitrage opportunities under various market conditions. Three types of agents, including profit-maximizing arbitrageurs, traders with exogenous motives and liquidity providers (both active/Markowitz portfolio optimizing and passive), interact in the Uniswap and a stochastic reference markets. The results show that Uniswap tracks market prices closely in different market environments, and Constant Product Curves have the potential to be price oracles [3].

3   Background - AMM Curves and Key Metrics

Consider a liquidity pool with two coins, whose amounts are denoted by x and y. For convenience, we will refer to these two tokens as X and Y. The AMM will allow the exchange of one token for another following a given function f as follows:

$$y = f(x) \tag{1}$$
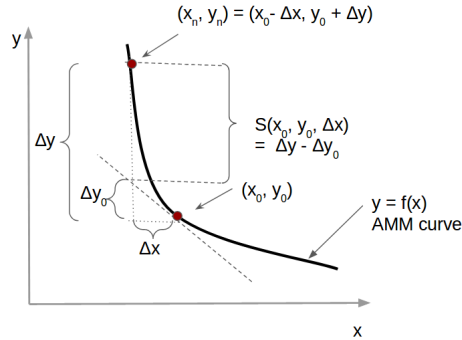
Figure 1 Illustration of Price Slippage on a Trade.

We refer to a plot of this function showing all allowed combinations of y and x as the *AMM curve*. For example, there can be a constant product curve, which is $y = \frac{k}{x}$ or a constant sum curve, which would be denoted as $y = c - x$. It is generally considered reasonable for the AMM curve to be convex and monotonically decreasing because this ensures (as we shall see in the next section) that the price for the token X is monotonically decreasing as a function of its availability in the pool, as should be expected of a typical supply curve.

3.1 Price Curve

Given an AMM curve, we can derive the price of the X token as follows:

$$p_X(x, y) = -\frac{dy}{dx} \tag{2}$$

For example, for the constant product curve, we would get $p_X(x, y) = \frac{k}{x^2}$ and likewise for the constant sum curve, we would get that $p_X(x, y) = 1$.

A plot of $p_X(x, y)$ versus x shows how the price of token X varies with its supply in the liquidity pool. Such a curve is referred to as a *price curve*. Note that if $f(x)$ is monotonically decreasing, then the price will always be positive, and if $f(x)$ is convex, then the price curve will be monotonically decreasing (as it should, being a type of supply curve).

3.2 Value of the pool

Given the definition of price, we can also assess the value of a given liquidity pool (measured in terms of Y) as follows:

$$V_p(x, y) = p_X \cdot x + y \tag{3}$$

3.3 Slippage

For curve-based AMM, slippage is defined as the loss incurred by a trader due to the price mismatch between the pool price at which the trade is initiated and the effective price obtained during the trade. Let us consider a trader seeking to buy Δx units of token X when the LP is at a state (x_0, y_0). Say that on the curve, the new point after the trade will be (x_n, y_n), where $x_n = x_0 - \Delta x$. The amount that the trader would then need to put into the LP will be $\Delta y = y_n - y_0$. If the pool price at the original point was p_0, then the buyer would have to pay $\Delta y_0 = p_0 \Delta x$. The difference between Δy and Δy_0 is defined as the slippage loss $S(x_0, y_0, \Delta x)$. This is illustrated in Figure 1.

Similarly, when the trader wishes to sell Δx units of token X, the gap between the $\Delta y_0 = p_0 \Delta x$ that the trader would like to receive and the $\Delta y = y_0 - y_n$ that he will actually receive would be the slippage loss on the sale, which could be expressed as $S(x_0, y_0, -\Delta x)$.

It is easy to see that on a constant sum AMM, the slippage loss is always 0 (because the price is constant at all points on the curve, or, equivalently, the tangent line at any point and the AMM curve always coincide). On any strictly convex curve, because the tangent line is always below the curve, the slippage loss will always be a positive quantity (i.e., the trader always incurs a penalty). The total slippage will be higher for a larger trade, and therefore acts as a disincentive for a trader to make large trades with the LP.

3.4   Divergence Loss

In general, when a trade is made, the price may change, as the original pair of values (x_o, y_o) moves to a new pair (x_n, y_n) following the curve, resulting in a new price p_n. Accordingly, the value of the liquidity pool could potentially decrease after a trade. This decrease as a relative or percentage decrease is referred to as divergence loss δ, and can be formally defined as follows:

$$\delta = \frac{V_{p_n}(x_n, y_n) - V_{p_n}(x_0, y_0)}{V_{p_n}(x_0, y_0)} \tag{4}$$

We work out below the divergence loss for the two example curves.

3.4.1   Divergence loss for constant-product curve

For the constant product curve, recall that the following hold:

$$p_n(x_n, y_n) = \frac{k}{x_n^2} \implies x_n = \sqrt{\frac{k}{p_n}} \tag{5}$$

$$y_n = \frac{k}{x_n} \implies y_n = \sqrt{k \cdot p_n} \tag{6}$$

Similarly, we also have that $x_o = \sqrt{\frac{k}{p_o}}$ and $y_o = \sqrt{k \cdot p_o}$.

Then we can define $V_{p_n}(x_n, y_n)$ as follows:

$$V_{p_n}(x_n, y_n) = p_n \cdot x_n + y_n$$
$$= p_n \sqrt{\frac{k}{p_n}} + \sqrt{k \cdot p_n}$$
$$= 2\sqrt{k \cdot p_n} \tag{7}$$

Likewise, we can define $V_{p_n}(x_o, y_o)$ as follows:

$$V_{p_n}(x_o, y_o) = p_n \cdot x_o + y_o$$
$$= p_n \sqrt{\frac{k}{p_o}} + \sqrt{k \cdot p_o} \tag{8}$$

Based on the above two equations, we can calculate the divergence loss as follows:

$$\delta = \frac{2\sqrt{k \cdot p_n} - p_n \sqrt{\frac{k}{p_o}} + \sqrt{k \cdot p_o}}{p_n \sqrt{\frac{k}{p_o}} + \sqrt{k \cdot p_o}} \tag{9}$$

Denoting by ρ the ratio of the two prices $\frac{p_n}{p_o}$, the divergence loss for the constant product curve can be simplified to:

$$\delta = \frac{2\sqrt{\rho} - 1 - \rho}{1 + \rho} \tag{10}$$

This result is given in [17].

3.4.2 Divergence loss for constant-sum curve

Here the price is always 1. The two values can be written as follows:

$$V_{p_n}(x_o, y_o) = x_o + y_o = c$$
$$V_{p_n}(x_n, y_n) = x_n + y_n = c \tag{11}$$

Since both are the same, the liquidity pool does not show any change in value, and thus the divergence loss in this case will be 0.

4 Dynamic curves

In the prior work on AMMs, the curve has a fixed form and the exact shape is determined by the initial total liquidity. E.g., in the constant product curve, the parameter $k = x_i \cdot y_i$ where x_i, y_i are the initial amounts of the two tokens. In other words, the curve can only change if the liquidity providers add/remove tokens from the pool, but not from trading activity.

Consider a trade that happens while the market price of token X remains unchanged at some price p_{mkt}. If the pool changes from the state (x_o, y_o) to a new state (x_n, y_n), then the pool price would potentially change from $p(x_o, y_o)$ to $p(x_n, y_n)$ (assuming the curve is not the constant-sum curve in which case there is no change in the pool price). This can result in at least a temporary difference between the pool price and the market price. As we do in the rest of the paper, we are assuming here that the pool's capitalization is a relatively small fraction of the total market capitalization of the underlying assets so that the market price is not determined or affected by the pool price.

Another reason for a temporary difference between the pool price and the market price could be that the market price changes due to some external market conditions. In either case, traditionally, it is expected that these temporary differences will be erased by the action of arbitrageurs, restoring the pool price back to the market price.

We propose a new mechanism that instead changes the curve every time the market price changes in such a way as to ensure that the current pool price will always equal the market price, *without requiring action by external arbitrageurs*. We illustrate below how this new mechanism would generalize the constant-product and constant-sum curves – the same approach can be used to generalize other smooth, decreasing, convex curves to the dynamic setting as well.

4.1 Dynamic curve adjustment to generalize constant-sum

In this case, we can describe the market-price-tracking dynamic curve as follows:

$$p_{mkt}(t) \cdot (x(t) - a(t)) + y(t) = c \tag{12}$$

Here, the parameter $a(t)$ will also be adjusted dynamically when the market price changes, to ensure that the new linear curve passes through the current pair of $(x(t), y(t))$ values. For simplicity, say the market is initialized at some pair $(x(0), y(0))$ at a market price of 1. Then c could be set to be $x(0) + y(0)$, with the original $a(0) = 0$.

If the market shifts to a price of $p_{mkt}(t)$ at some time t and the liquidity pool at this arbitrary time is $(x(t), y(t))$, then the value of $a(t)$ will also be adjusted as follows to match the above dynamic curve:

$$a(t) = x(t) - \frac{c - y(t)}{p_{mkt}(t)} \tag{13}$$

Intuitively, this dynamic curve is always a line that has the slope corresponding to the current market price and always passing through the current liqudity pair $(x(t), y(t))$.

Any trade that happens uses the current (instantaneous) curve. This allows the constant sum AMM to flexibly support a wider range of market prices while still providing 0 slippage compared to the original design (which allows only a fixed pool price and thus will not work when the market price is dramatically different).

4.2   Dynamic curve adjustment to generalize constant-product

In this case, we can describe the market-price-tracking dynamic curve as follows:

$$w(t) \cdot (x(t) - a(t)) \cdot y(t) = k \tag{14}$$

Or alternatively, as:

$$y(t) = \frac{\frac{k}{w(t)}}{x(t) - a(t)} \tag{15}$$

Note that in the above expressions, $x(t)$ and $y(t)$ must always be strictly positive; $a(t)$ must be constrained to be always strictly less than $x(t)$; and $w(t)$ should always be strictly positive. The instantaneous price corresponding to the dynamic version of the constant product curve can be defined as follows:

$$p_X(t) = \frac{k}{w(t)} \cdot \frac{1}{(x - a(t))^2} \tag{16}$$

When the market price changes, then both $w(t)$ and $a(t)$ will have to be changed in order to (a) make sure that the new market price $p_{mkt}(t)$ matches $p_X(t)$ in equation (16) and (b) x(t),y(t) match the curve described in equation (14). Thus we have to solve two equations and two unknowns. The solution turns out to be the following :

$$a(t) = x(t) - \frac{y(t)}{p_{mkt}(t)}$$
$$w(t) = \frac{k \cdot p_{mkt}(t)}{y(t)^2} \tag{17}$$

We remark: the first expression above ensures the requirement mentioned above that $a(t)$ will remain strictly less than $x(t)$ and the second expression ensures that $w(t)$ is strictly positive, so long as k, $p_{mkt}(t)$, $x(t)$ and $y(t)$ are all kept strictly positive at all.

4.3   From divergence loss to slippage gain

As with the static setting, there is no slippage loss for traders or divergence loss in the case of dynamic constant sum AMM. This is because in the absence of any change in market price, the pool price does not change during a trade.

In the case of the dynamic constant product AMM, the traders do experience a slippage loss just like in the static constant product AMM case. However, corresponding gain in value is accrued entirely to the liquidity pool and could be referred to as a *slippage gain* for the LP. Further, in the absence of change in the market price, because the pool price does not change in the dynamic constant product curve, there is no divergence loss. Rather, the LP benefits from each trade by the same slippage gain. Thus, the dynamic constant product AMM provides a strict improvement from the LP's perspective. This result, in fact, generalizes to the dynamic version of any strictly convex curve, as we show below.

Proposition 1. *In a dynamic AMM based on a family of monotonically decreasing $y = f(x)$ curves that are strictly convex, when the market price remains fixed, the LP will gain value after each trade by an amount equivalent to the slippage loss of the trader.*

Proof. Assuming the market price does not change during a trade, for any strictly convex curve, the trader suffers a slippage loss at each trade. This is because the tangent to the curve (whose slope is equal to the pool price and therefore the market price) lies below the curve if it is strictly convex. If the trader buys X tokens, it will therefore have to give the pool an amount of Y tokens that exceeds what it should have given at the current market price. Likewise, if the trader sells X tokens, it will receive an amount of Y tokens less than what it should have received at the current market price. The gap, in either case, corresponds to the slippage loss. An equivalent amount is gained by the LP (when the trader buys X tokens, the excess Y tokens are sent to the LP; when the trader sells X tokens, the gap corresponds to Y tokens are withheld by the LP). There is no other source of divergence loss for the LP because the pool price is readjusted to the market price immediately after execution of the trade - its value increases precisely by the amount of slippage loss experienced by the trader. ◀
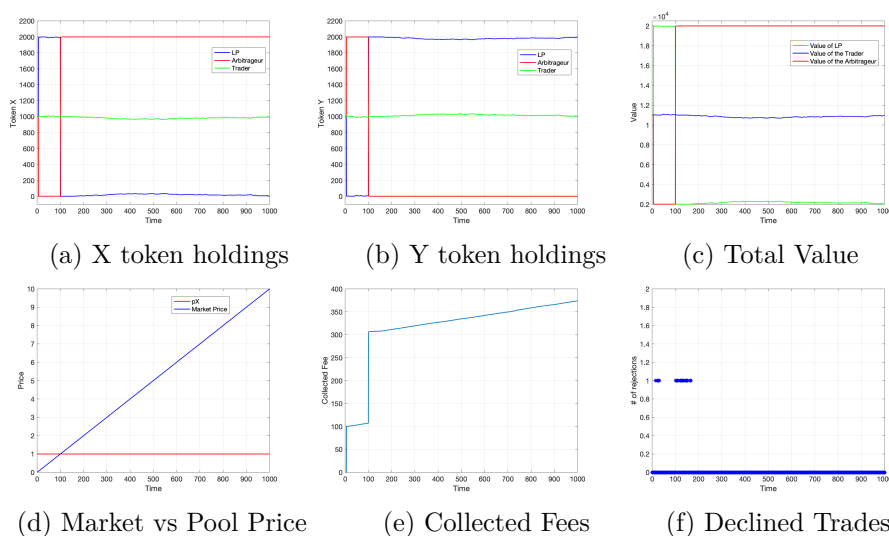
5 Simulation Results

We conduct a number of simulations to compare the performance of four different AMM's, two static AMM's (Constant-Sum AMM, Constant-Product AMM) and their two dynamic generalizations that we have introduced in section 4. Although it is not practically implemented due to its shortcomings, we nevertheless use the constant-sum AMM as a baseline to highlight how the dynamic version of this scheme has advantages and may be more interesting from a practical perspective.

5.1 Simulation Setup

In all our simulations, we consider the three parties involved:
- The liquidity pool (LP).
- A trader (which could be viewed as a collection or series of independent traders who are merely interested in exchanging a relatively small amount of one of the tokens in the LP for another).
- An arbitrageur (a profit-maximizing agent responding specifically to any gaps in price between the pool and the broader market; the arbitrageur in our simulation could also be viewed as a collection or series of independent arbitragers).

Initially, all three parties have 1000 X tokens and 1000 Y tokens. At each time, a trader makes a random trade (buy or sell X token) drawn from a standard normal distribution (i.i.d. Gaussian process with zero mean and unit variance). The relatively small standard

(a) X token holdings   (b) Y token holdings   (c) Total Value

(d) Market vs Pool Price   (e) Collected Fees   (f) Declined Trades

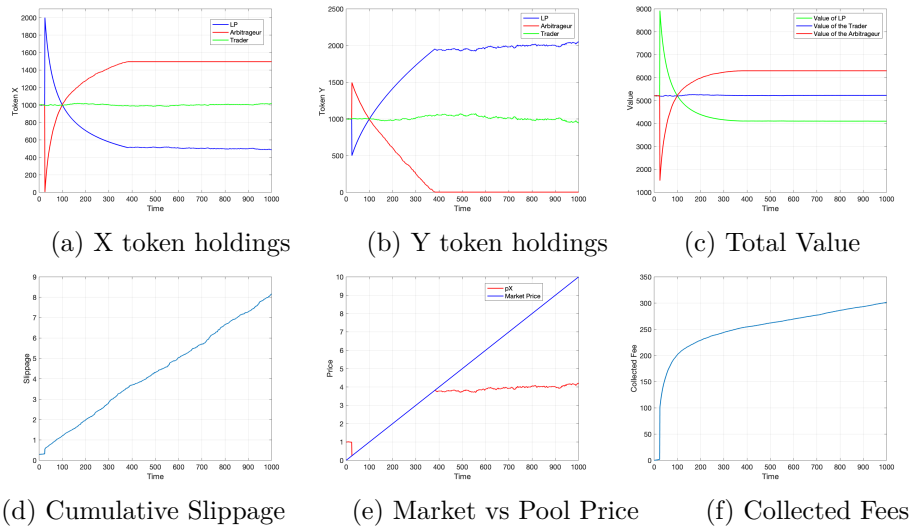**Figure 2** Simulation of a Static Constant Sum AMM.

deviation would result in relatively light trading or profit from transaction fees collected from trades (that are not for the purpose of arbitrage). We use the exact same sequence of random trades in evaluating all four AMM's, to enable a fair comparison. We model the market price as undergoing a linear increase from 0 to 10 over the course of 1000 time steps.

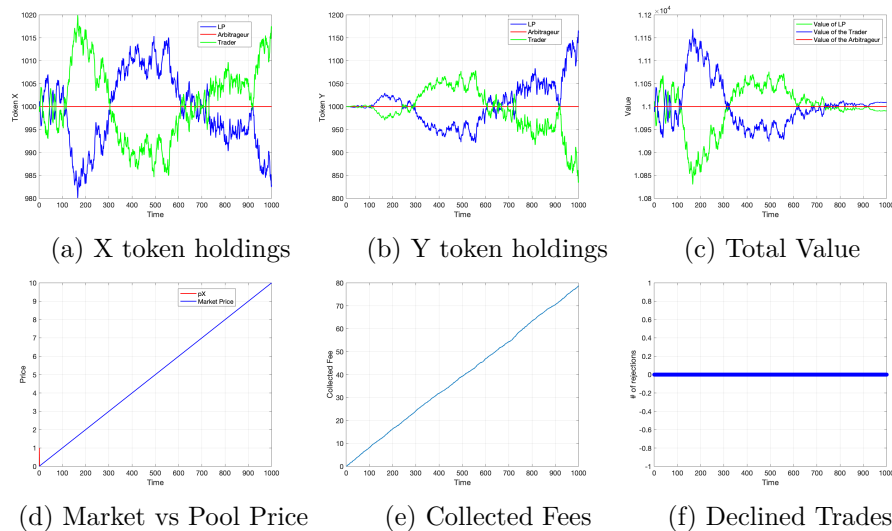## 5.2 Static Constant Sum AMM

Figure 2 shows the results for the baseline static constant AMM. As shown by Figure 2(d), the market price is initially below the pool price of 1, and after 100 time steps it switches to being above the pool price. As can be seen from Figure 2 (a), (b), the arbitrageur initially sells all its X tokens to buy up all of the Y tokens from the LP when the market price is low, and then it buys up all the X tokens from the LP when the market price becomes high. By draining the pool of its liquidity in one of its assets, to the pool's disadvantage, the arbitrageur effectively extracts most of the total value from the LP. Figure 2(c) shows the total value of the LP, the arbitrageur and the trader measured in terms of the market price at the end of the simulation (when X tokens are worth 10 Y tokens); it shows that the LP ends up with only about a tenth of the value it had at the beginning of the simulation. The LP does collect some fees from the arbitrageur (per Figure2(e)), but they are relatively modest compared to the loss in value. Although the trader doesn't suffer slippage loss in this case, it can be disappointed and face a rejection of its requested trade whenever the pool lacks sufficient tokens of the type sought by the trader (as shown in figure 2(f); this happens particularly early on when the arbitrageur is able to cause the LP to be nearly empty of one of its assets, then another, disrupting the LP's ability to serve the regular trader).

5.3 Static Constant Product AMM

Figure 3 shows the results for the static constant product AMM. Here too, we can see from Figures 3(a) and (b), the arbitrageur has a big impact on the token holdings of the LP. Over time, as the market price goes up, the arbitrageur keeps buying X tokens from the pool, equalizing the pool price and the market price. Eventually the arbitrageur has no more Y tokens available (after which point the pool prize stays below the market price, see

(a) X token holdings   (b) Y token holdings   (c) Total Value

(d) Cumulative Slippage   (e) Market vs Pool Price   (f) Collected Fees

Figure 3 Simulation of a Static Constant Product AMM.

(a) X token holdings   (b) Y token holdings   (c) Total Value

(d) Market vs Pool Price   (e) Collected Fees   (f) Declined Trades

Figure 4 Simulation of a Dynamic Constant Sum AMM.

Figure 3(e)). As shown in Figure 3(c), the total value of the LP reduces while that of the arbitrageur grows until the latter runs out of Y tokens. In reality, however, a well-funded arbitrageur would keep going until all X tokens are depleted from the LP, causing it to lose even more value. The trader suffers some constant slippage on average in every trade, as illustrated by the cumulative slippage showing a linear increasing trend in Figure 3(d). There are again some collected fees from the arbitrageur and the regular trader, but here too they pale in comparison to the loss in value caused by the arbitrageur's actions. It is clear that just like with the constant sum setting, in the long term, the LP is at the mercy of the arbitrageur in the presence of significant market price volatility.
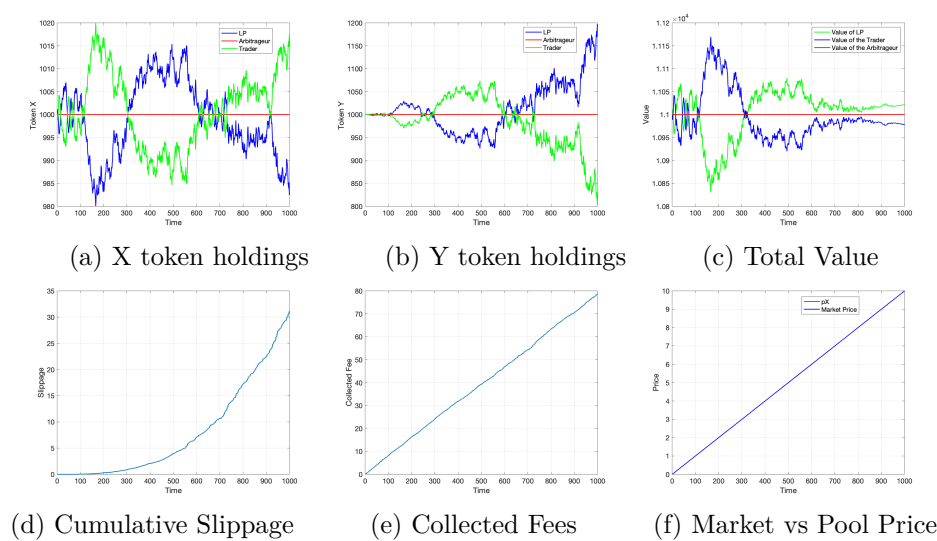
Figure 5 Simulation of a Dynamic Constant Product AMM.

5.4   Dynamic Constant Sum AMM

Figure 4 shows the results for the dynamic constant sum AMM. Here the arbitrageur is eliminated as the pool price is automatically adjusted to the market price after each trade. The total tokens of either types X or Y remain the same and the holdings of the LP and the trader are mirror images of each other, as shown in Figures 4(a) and 4(b). The total value of the LP doesn't show a dramatic change other than random fluctuation caused by stochastic trading on the pool. In the absence of aggressive arbitrage, the collected fees are somewhat modest. As shown by Figure 4(e), the LP collects the expected amount of fees (about 80 over 1000 trades, corresponding to the expectation of the absolute value of a normal random variable). A noticeable fact about the dynamic constant sum AMM is that *the LP maintains more than 85% of the original amount of tokens of both types at all times* during the simulation. This is in sharp contrast to the liquidity problems observed in the static settings (Figures 2 and 3). As a result, there are no declined trades, as evidenced by Figure 4(f). Another related comment to be made here is that there is little incentive for large trades in this dynamic AMM because the price is always pegged to the market price and the transaction fees (being a constant percentage) impose a higher cost in absolute terms on larger trades.

5.5   Dynamic Constant Product AMM

Finally, Figure 5 shows the results for the dynamic constant product AMM. Here too, the arbitrageur is eliminated as the pool price is automatically adjusted to the market price after each trade (see Figure 5(f)), and the X and Y tokens held by the trader and LP mirror each other as shown in Figure 5(a) and Figure 5(b). The total value of the LP in Figure 5(c) doesn't show a dramatic change other than the fluctuation caused by stochastic trading on the pool and a slight increase in value for the LP due to the slippage gain discussed earlier in this paper. The corresponding cumulative slippage loss for the trader, equivalent to the cumulative slippage gain for the LP, is shown in Figure 5(d), and it is this gain that results in a slight positive drift in the LP value over time. As with the dynamic constant sum AMM, in the absence of aggressive arbitrage, the LP collects the expected amount of fees from the

regular trader (≈ 80 over 1000 trades, corresponding to the expectation of the absolute value of a normal random variable). Compared to the dynamic constant sum setting, here, there is even less incentive for any trader to execute a large trade as they would suffer a large slippage loss in addition to paying transaction fees. Figure 5(a) and Figure 5(b) show that the LP maintains high liquidity in both assets again – more than 90% of the original amount of tokens of both types at all times during the simulation.

6 Conclusions

We have given a detailed introduction to curve-based AMMs for decentralized cryptocurrency exchanges in this work. We introduced a new approach to operating such curve-based AMM decentralized exchanges that utilizes an oracle with a real-time market price feed to continuously and automatically adjust the pool price to the market price by dynamically adjusting the curves over time. We showed that in such a dynamic AMM, there is no room for arbitrage. The slippage loss for traders is converted to an equivalent gain for the liquidity pool. The LP maintains a high level of liquidity in all assets and its total value stays fairly stable over time. Such a dynamic AMM results in a slightly lower collection of transaction fees due to the elimination of arbitrageurs, but this loss is more than offset by the ability to maintain the total value of the pool.

From a practical perspective, implementing such a dynamic AMM requires the use of a low-latency and accurate market price oracle. It would be of great interest to develop and evaluate a real-world decentralized exchange based on this approach.

References

1   Hayden Adams, Noah Zinsmeister, and Dan Robinson. Uniswap v2 core, 2020. URL: `https://uniswap.org/whitepaper.pdf`.

2   Guillermo Angeris and Tarun Chitra. Improved price oracles: Constant function market makers. *arXiv preprint*, 2020. `arXiv:2003.10001`.

3   Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of uniswap markets. *Cryptoeconomic Systems Journal*, 2019.

4   Bancor. Announcing Bancor V2. URL: `https://blog.bancor.network/announcing-bancor-v2-2f56b515e9d8`, November 2020. Accessed: 2020-12-18.

5   Dmitriy Berenzon. Constant function market makers: Defi's "zero to one" innovation. URL: `https://medium.com/bollinger-investment-group/constant-function-market-makers-defis-zero-to-one-innovation-968f77022159`, May 2020. Accessed: 2020-12-18.

6   Vitalik Buterin. Let's run on-chain decentralized exchanges the way we run prediction markets. URL: `https://www.reddit.com/r/ethereum/comments/55m04x/lets_run_onchain_decentralized_exchanges_the_way/`, October 2016. Accessed: 2020-12-18.

7   Chainlink. How to bring more capital and less risk to automated market maker dexs. URL: `https://blog.chain.link/challenges-in-defi-how-to-bring-more-capital-and-less-risk-to-automated-market-maker-dexs/`, June 2020. Accessed: 2020-12-18.

8   Chainsecurity. Upcoming exchange concepts – a farewell to the order book? URL: `https://chainsecurity.com/disruption-or-disillusion/`. Accessed: 2020-12-18.

9   Richard Chen. A comparison of decentralized exchange designs. URL: `https://thecontrol.co/a-comparison-of-decentralized-exchange-designs-1deef249f56a`, April 2019. Accessed: 2020-12-18.

10 CoinMarketCap. Cryptocurrency prices, charts and market capitalizations. URL: `https://coinmarketcap.com/`. Accessed: 2020-12-18.

11 Michael Egorov. Stableswap – efficient mechanism for stablecoin liquidity. URL: `https://curve.fi/files/stableswap-paper.pdf`, 2019.

12 Gnosis. White paper. URL: `https://github.com/gnosis/research/blob/master/gnosis-whitepaper.pdf`, 2017.

13 Robin Hanson. Logarithmic markets scoring rules for modular combinatorial information aggregation. *The Journal of Prediction Markets*, 1(1):3–15, 2007.

14 Eyal Hertzog, Guy Benartzi, and Galia Benartzi. Bancor protocol: Continuous liquidity for cryptographic tokens through their smart contracts. *Bancor White Paper*, 2018.

15 Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Online article*, 2019.

16 Jack Peterson, Joseph Krug, Micah Zoltu, Austin K Williams, and Stephanie Alexander. Augur: a decentralized oracle and prediction market platform. *arXiv preprint*, 2015. `arXiv:1501.01042`.

17 Pintail. Uniswap: A good deal for liquidity providers? URL: `https://pintail.medium.com/uniswap-a-good-deal-for-liquidity-providers-104c0b6816f2`, August 2020. Accessed: 2020-12-18.

18 Totle. Do dexs have a slippage problem? URL: `https://medium.com/totle/do-dexs-have-a-slippage-problem-68c3a4fe5161`, September 2019. Accessed: 2020-12-18.

19 Yongge Wang. Automated market makers for decentralized finance (defi). *arXiv preprint*, 2020. `arXiv:2009.01676`.

20 Will Warren and Amir Bandeali. 0x: An open protocol for decentralized exchange on the ethereum blockchain, 2017. URL: `https://github.com/0xProject/whitepaper`.

21 Lawrence H White. The market for cryptocurrencies. *Cato J.*, 35:383, 2015.