Title

The Ultimate Curve: An Automated Market Maker with a Linear Stretch

Author

Huan, Jason Yu

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

The Ultimate Curve:
An Automated Market Maker
with a Linear Stretch

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Computer Science

by

Jason Yu Huan

2024

ABSTRACT OF THE THESIS

The Ultimate Curve: An Automated Market Maker with a Linear Stretch

By

Jason Yu Huan

Master of Science in Computer Science

University of California, Los Angeles, 2024

Professor Eliezer M. Gafni, Chair

An Automated Market Maker (AMM) is a smart contract created to allow the exchange of assets according to a mathematical function defined in the contract. As the exchanges occur, if they start to deplete one of the assets held by the contract, we would like to increase the relative price of the asset being depleted. On the other hand, when the balances of the assets are not too far from each other, we would like the relative price of the assets to stay as close to a constant as possible.

These two competing wishes were handled by creating a formula consisting of the weighted sum of two functions: one taking care of the neighborhood of the starting point and the other under the condition when the assets diverge significantly.

We propose an *ultimate* compromise: a straight line in the neighborhood, and the translated original function that treats all points on the function the same without a specific bias when the assets diverge.

We present in detail the case of two assets, and we propose (without analysis) a generalization of the construction to the multi-asset case.

The thesis of Jason Yu Huan is approved.

Amit Sahai

Leonard Kleinrock

Eliezer M. Gafni, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGMENTS

I would like to acknowledge the Stellar Development Foundation for providing me with a substantial gift and give thanks to the support they gave me in pursuing my master's degree studies at UCLA.

I would also like to thank the generous assistance of Professor Leonard Kleinrock, who gave me his gracious support in the start of my graduate degree and the funding for my first quarter of graduate school.

Lastly, I would like to thank everyone who helped me so that I was able to focus on my studies full-time without the distraction of the want for money.

Introduction

Uniswap [1], introduced around 2019, is a smart contract-based automated market maker (AMM) that allows traders to swap between a pair of coins (X, Y) held inside a smart contract. It consists of a function, where in the case of Uniswap, the constant product function is $X \times Y = K$ for some constant K determined by the amount of X and Y tokens the contract holds.

A *Liquidity Provider* changes K by depositing X and Y into the contract (and accumulating fees from the traders); the effect of the traders only causes movement along the function and does not change the function itself, namely the invariant K which remains the same from the start to finish of all swaps. Note that the constant-product function is unique in the fact that its derivative is equivalent to the ratio of X and Y on that point of the curve, allowing anybody to inherently find the price by dividing the balances of the tokens in the AMM together.

The reason a mathematical function is used as a trading algorithm is due to the high transaction costs and latency of the Ethereum blockchain, of which Uniswap was originally launched on. Previous attempts at smart contract-based platforms for decentralized exchange such as EtherDelta used a central order limit book model, which does poorly in the presence of high latency and high transaction fees, since each individual limit order (e.g. BUY at 5.4 token

X per token Y) must be created or selected, and the average time for a block in Ethereum is roughly 13.3 seconds in the proof-of-work model. The result of this was many orders being too expensive or failing to execute resulting from collisions with others, and thus the main source of liquidity for tokens became centralized exchanges (rather than distributed smart contracts) such as Coinbase or Binance.

In the Ethereum blockchain, tokens are represented as either the native Ether token (ETH), or an ERC-20 interface which standardizes the functionality of the token as a smart contract. When transferred into another smart contract or user's account, the ERC-20 contract updates its balances record in order to decrease the balance of the sending account and increase the balance of the receiving account accordingly.

Using the constant-product market making algorithm, Uniswap was able to reach over $10 billion dollars worth of liquidity deposited into their protocol in the market-leading record year of 2021. Before the advent of Uniswap, most users did not have any way to exchange their tokens issued on the Ethereum blockchain without going to centralized exchanges. With the advent of Uniswap, the field of decentralized exchanges (DEXs) was born and with it, decentralized finance.

Following the release of Uniswap, Curve introduced its StableSwap whitepaper[2] which presents itself as an efficient liquidity mechanism for stablecoins, which are price-stable tokens pegged to a fiat currency. As there are many different offerings of stablecoins, swapping between any two identically-backed stablecoins should not involve any significant price disparity between the two. As a result, the constant-product market making mechanism is blended with a constant-sum market making mechanism definable as $X + Y = K$, where the function $Y = f(X)$ with derivative $dY = -dX$ offers token swaps at a one-to-one exchange ratio and thus blended with the constant-product invariant allows for much more liquidity at ranges near a single price than in the traditional constant-product market maker.

The venue of Curve quickly became the leading decentralized exchange for stablecoins, as its AMM reached roughly $25 billion dollars of total liquidity deposited at its peak in 2022. The result of a lower price difference between the spot price and the actual price when executing trades with some input size, also known as *slippage*, led Curve to creating a large market for stablecoin exchanges.

Likewise, in the direction of multiple assets being traded against one another, the introduction of the mechanism in Balancer[3] generalized the constant-product market maker to multiple dimensions, which allowed for trading tokens in a "pool" of multiple assets. This invention allowed for *weights* of the pool such that each asset composes a certain percentage of the trading pool. This mechanism not only opened the door to multiple assets in a pool, but begs the question for other pegged-asset liquidity with such an aspect for improvement.

Given the motivation by Curve's StableSwap to allow for a nearly 1-to-1 exchange ratio, we now create a market maker that allows for any arbitrary spot price q such that there is a linear stretch in the market-making curve which transitions back into the shape of the constant-product curve while outside the linear stretch. This linear stretch allows for an invariant of $qX + Y = K$ for token swaps, departing from the need to provide constant-sum style liquidity only between stablecoins and giving the option to include it between differing-valued token pairs such as those who have a constant spot price against one another.

In our contribution, we introduce a novel mechanism where the market-making curve is completely linear for a stretch of the curve, and then shifts back to the shape of the constant-product curve beyond that range. This allows for a constant price at a concrete range of liquidity, while transitioning and gaining back the properties of the constant-product market maker when beyond the range, importantly a defined price for all possible balances between the tokens. This is accomplished through splitting the normal constant-product curve into two sections, where the derivative at the splitting point is equivalent to the price desired over the linear stretch, and then translating the two sections outwards horizontally and vertically according to a parameter Δx to create a smooth transition back to the constant-product shaped curve.

We then begin to explore a generalization of this mechanism to multiple dimensions, wherein the curve forms a simplex in the linear region with slopes in each dimension given by the prices desired, and the transformed multidimensional invariants are created through radially-outward translations of the constant-product invariant, without giving analysis on the differentiability of the invariant in multiple dimensions.

Motivation

We intrinsically desire three properties for our AMM design. First, we would like the function to be continuously differentiable at all points, and for its derivative to also be continuously differentiable. This property of twice-differentiability allows for a continuous price function along any possible ratio of balances offered by the AMM. This property also minimizes the possibility of an arbitrage opportunity being created in usage due to the AMM itself.

Secondly, we aim for the range of the derivative function $f'(X)$ such that $0 < f'(X) < \infty$ so as to offer any price possible depending on extrinsic market prices, where the price in the AMM is able to come in parity with the price of the exogenous market due to some arbitrage available between the two. Any arbitrageur should be freely available to trade between this AMM and an external market source, with a possible price difference between the two being quickly closed such that the difference approaches some margin of arbitrage cost.

Third, we would like the AMM to be convex so that the first derivative as the price function is monotonically increasing along the entire domain. This property ensures that the exchange rate of tokens received in return decreases as a function of the amount input. This property is desirable as it allows the AMM to find the global optimal price between the two tokens without having to set any conditions on the size of its trades.

The goal of such an AMM with a linear stretch is to allow for the qualities of the properties stated whilst also being an efficient market-maker for stablecoins and other constant-pegged assets, since they should not deviate beyond a specific price between each other over a longer period of time. In the case where such an AMM becomes the dominant source of liquidity, the arbitrage available on those pegged assets traded should become larger than what is available on existing venues and theoretically lead to a tighter peg overall for all the assets included.

Spot Price

Akin to the constant spot-price of a constant-sum invariant, our mechanism allows a constant spot-price q between assets X and Y on the curve for the linear stretch between them. The defintion of the spot-price q at a given point $(x_p, f(x_p))$ on our function is defined as the infinitesimal exchange rate between the two tokens at that point expressed as a scalar value where $q \cdot x_p = f(x_p)$, which is equal to the negative of the first derivative of our function:

$$q = -f'(x_p) \tag{3.1}$$

In the case that we are looking to provide liquidity between identically-pegged stablecoins, we will pick our spot price to be one-to-one between them akin to a constant-sum invariant, i.e. $q = 1$. Using this spot price, we are able to create the market making function with a linear stretch with our slope desired using the spot price.

For those assets who are constantly-priced against one another but not necessarily equal to $q = 1$, the market making function is also able to change the spot price as desired. As a result, the liquidity providers are able to determine at which price and for how much amount a token is to be traded for against its trading pair asset, similar to a buy and a sell limit order. This limit order-like functionality is built into the AMM and offers precise liquidity at a constant price for traders, a novel feature for AMMs which is commonly offered in other asset exchange venues.

Transformed Invariant

To create our market making function, we first create a vector outwards from the origin to point $P = (x_p, f(x_p))$ on our function where $\frac{f(x_p)}{x_p}$ is equal to our desired spot price, q. Next, we choose how much proportional liquidity we would like to offer of our asset X at a constant spot price over the linear stretch, and take an accordingly "shortened" point along this vector P', such that $P' = (x_{min}, y_{min})$ where $x_{min} + \Delta x = x_p$. This will be our spot to begin our translation, and we will treat this point that lies on the curve of

$$K' = y_{min} \cdot x_{min} \tag{4.1}$$

as our *virtual liquidity*, where our market making function will behave as if it has a translated amount of this amount of liquidity when outside the linear stretch on the transformed curve, while offering a spot price exchange rate q while inside the linear stretch on the transformed curve. As a result, the spot price on the linear stretch is smooth with respect to the translation points from the virtual liquidity in equation 4.1.

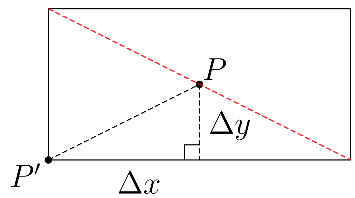

Figure 4.1: Linear stretch alongside horizontal and vertical translation

Pictured above in Figure 4.1 is the translation from our virtual liquidity to the liquidity offered by our AMM which includes the linear stretch. Note that Δy is simply a function of Δx multiplied by the spot price q at point P.

We next take the constant-product invariant from Uniswap where our base invariant is:

$$f(x) = \frac{K}{x} \tag{4.2}$$

And our transformed invariant is:

$$f(x) = \begin{cases} \frac{K-\Delta K}{x} + 2\Delta y, & x < x_{min} \\ \frac{K}{x_{min}} + 2\Delta y - q \cdot x, & x_{min} \leq x \leq x_{min} + 2\Delta x \\ \frac{K}{x-2\Delta x}, & x > x_{min} + 2\Delta x \end{cases} \tag{4.3}$$

Where the translated portion of the curve K sits above the virtual liquidity curve of K' by a vertical translation of $2\Delta y$ when $x < x_{min}$, and to the right of K' by a horizontal translation of $2\Delta x$ when $x > x_{min} + 2\Delta x$. The translated virtual liquidity is provided to the traders when the function is outside of the linear stretch, and the spot price liquidity is offered while inside the linear stretch.

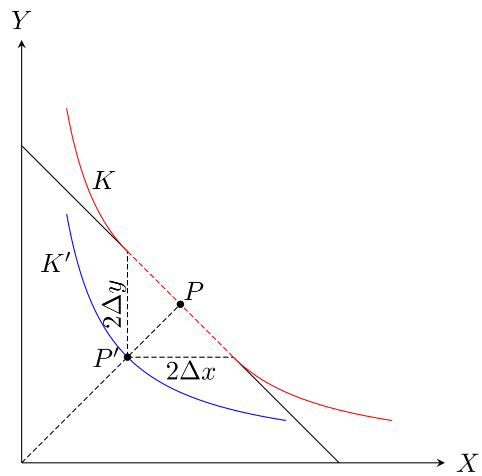

Figure 4.2: An example of a market making curve with $q = 1$

Pictured in Figure 4.2 is the constant-product invariant in blue, the constant-sum invariant in black, and our transformed invariant in red shown by values Δy and Δx. The example has spot price $q = 1$, with the virtual liquidity with point $P' = (2, 2)$ lying on the curve of $K' = 4$ and the point at the linear stretch $P = (3, 3)$ having an invariant of $K = 9$. The value $\Delta x = 1$ is given and thus $\frac{2\Delta x}{x_{min}+2\Delta x} = \frac{1}{2}$, or half of the asset X liquidity is provided on the linear stretch.

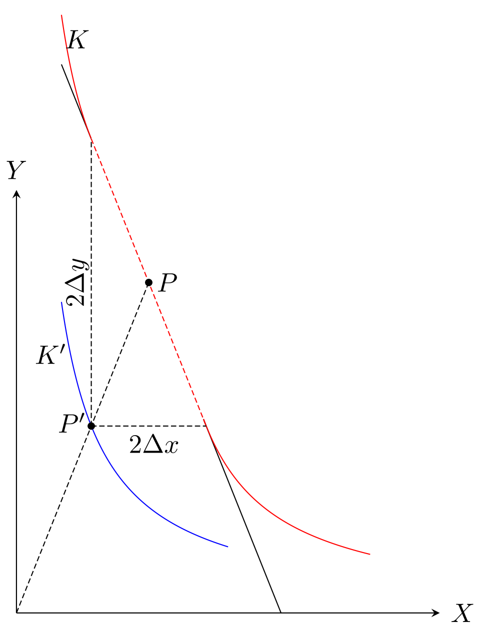

Figure 4.3: An example of a market making curve such that $q \neq 1$

In the preceding Figure 4.3 of the market making function, the translation has the value $\Delta x = 1.086$, with $q = 2.5$ as the spot price for asset X in terms of asset Y in the linear stretch. With $P' = (1.41, 3.54)$ and $P = (2.50, 6.25)$, the total proportion of liquidity r outside the linear stretch is $r = 0.404$, with the majority of the X asset liquidity being inside the linear stretch. This example highlights a market making function for two assets with a constant spot price between them that is not equal to $q = 1$.

Multidimensional Case

For the ease of explanation, we will only elaborate the three-dimensional case of three assets X, Y and Z and the normal constant-product function of $X \times Y \times Z = K$.

Consider the point of liquidity P of coordinates x, y and z. The point P determines the relative spot prices λ_x, λ_y and λ_z, satisfying $\lambda_x/\lambda_z = z/x$ and $\lambda_y/\lambda_z = z/y$ with z as our *numeraire* asset, and $\lambda_x + \lambda_y + \lambda_z = 1$.

As we did with the two dimensions, we consider our proportion r of reduced virtual liquidity P' such that $r = \frac{P'}{P}$ corresponding to x', y' and z' with $r = x'/x = y'/y = z'/z$ where $r < 1$, where we want to abandon the flat surface.

We now have four hyperplanes: the three basic hyperplanes $X > x'$, $Y > y'$ and $Z > z'$, and the hyperplane tangent to the constant product function $X \times Y \times Z$ at P, called F_T, which are all the positive-value points X, Y and Z satisfying the simplical invariant:

$$(X\lambda_x) + (Y\lambda_y) + (Z\lambda_z) = K \tag{5.1}$$

The intersection of these 4 half-spaces (for the tangent hyperplane, take the half space containing point P') define a 3 dimensional simplex. The faces of each corresponds to one hyper-plane. We consider the face defined by the tangent hyperplane F_T which (abusing notation) we will also call F_T. We consider its boundary which is a triangle which we call B_{F_T}.

We now define by construction the *ultimate curve* by radially "building the surface."

The first part of the surface is the full triangle F_T. This is the flat surface around the initial point of liquidity P. For each point p on the boundary of the triangle F_T, i.e. $p \in B_{F_T}$, we define a one dimensional function (of infinitely many): The three points P', P and p define the hyperplane T that contains them. The intersection of this hyperplane T with the surface formed by the virtual liquidity, the constant-product function $XYZ = K'$ for $K' = Kr$ (the surface that goes through P') gives us a one dimensional function going through P'. We divide the hyperplane into two half-hyperplanes by cutting it along the line defined by (P', P). We consider the part of the function which resides on the half containing p. We take this half-function and translate it from P' to p along the line of (P', p). Our surface now consists of the triangle F_T and all of the functions that we radially translated outward from the point of virtual liquidity P', each corresponding to some p on the boundary B_{F_T}.

Bibliography

[1] H. Adams, N. Zinsmeister, and D. Robinson. Uniswap v2 core. 2020.

[2] M. Egorov. Stableswap - efficient mechanism for stablecoin liquidity. 2019.

[3] F. Martinelli and N. Mushegian. A non-custodial portfolio manager, liquidity provider, and price sensor. 2019.