47th EBES Conference
"Eurasia Business and Economics Society"
April 18-20, 2024, Berlin, Germany

EXPLORING THE VERSATILITY OF BONDING CURVE MODELS IN DECENTRALIZED FINANCE: A HIGHLY SCALABLE MARKET MAKER MODEL FOR DIGITAL ASSETS

Alvis KAISER[1], Wurst TSAO[2], Patrick CAJINA CORTEZ[3]

[1] PhD of Economics, Burve Labs, Munich, Germany
[2] Master of Financial Mathematics, Burve Labs, Brussel, Belgium
[3] Core Contributor, Burve Labs, Barcelona, Spain
Email: info@burve.io

Abstract

This paper comprehensively explores the applications of bonding curve models in decentralized finance (DeFi), emphasizing token launching, unilateral liquidity management, and stablecoin trading. Using a mixed-methods approach, we integrate theoretical analysis with empirical experimentation. Mathematical modeling techniques elucidate bonding curve dynamics and their implications for various DeFi scenarios, while smart contract prototypes validate insights and assess practical feasibility. Our findings demonstrate the effectiveness of bonding curve models in facilitating fair token launches and market making without traditional liquidity pools. Moreover, algorithmic integration and optimization showcase the viability of unilateral liquidity provision using bonding curves, enabling dynamic adjustment of liquidity provision rates. Additionally, we introduce a novel stablecoin model leveraging dual bonding curves to enhance stability and capital efficiency. Bonding curve models hold promise for addressing liquidity challenges and promoting financial inclusion in decentralized ecosystems. Our study underscores their transformative potential in DeFi and emphasizes the need for further research and experimentation to fully harness their capabilities.

Keywords: bonding curve, decentralized finance, automated market maker, liquidity, stablecoin, blockchain

JEL Classification: G23, C61, E42

Introduction

Blockchain technology, alongside smart contracts and decentralized finance (DeFi), is assuming an increasingly significant role in the global economic ecosystem (Voshmgir, 2019). This emergence not only presents novel approaches but also introduces a plethora of opportunities for transforming traditional financial systems and fostering innovation in various sectors. Automated Market Makers (AMMs) play a pivotal role in decentralized finance (DeFi) and the broader cryptocurrency ecosystem. These algorithms enable decentralized exchanges to function seamlessly, providing liquidity, facilitating trading, and powering various financial applications (Pourpouneh et al., 2020). AMMs have become indispensable tools for traders, investors, and developers within the crypto space. Their significance extends beyond mere trading; they underpin decentralized lending platforms, yield farming protocols, and other innovative DeFi solutions, democratizing access to financial services and fostering a more inclusive global economy.

Common AMMs encompass order books and liquidity pools, each characterized by distinct advantages and drawbacks. In contrast, the bonding curve presents a novel approach independent of these traditional models. Originating from S. De la Rouviere's (2017) pioneering work, the bonding curve's theoretical foundation in token pricing, has found initial application in networks like Ethereum. This paper systematically advances the AMM theory of the bonding curve, asserting its status as a more Turing-complete automatic market maker option. For the first time, it comprehensively examines the architecture of the bonding curve as an automatic market maker model. Furthermore, it introduces and discusses in depth an optimization model for token launching, unilateral liquidity provision and management, and stablecoin trading within the context of the bonding curve.

1. Background and Literature Review

Schär (2021) and Pourpouneh et al., (2020) delineate the pros and cons of centralized and decentralized exchanges. While centralized exchanges offer ease of use, they expose traders to risks such as loss of asset custody and susceptibility to security threats. In contrast, decentralized exchanges (DEXs) mitigate these risks by eliminating the need for trust in a single entity and allowing traders to retain control of their assets through smart contracts (Xu et al., 2021).

In a centralized exchange, the automated market maker is the order book, while in decentralized exchanges, liquidity pools serve as the automated market maker (AMM), enabling token owners to provide liquidity for token swaps (Mohan, 2022). Various subtypes of liquidity pool AMMs exist, including Constant Product Market Maker (CPMM) pioneered by Uniswap (Adams et al., 2021), Constant Mean Market Maker (CMMM) exemplified by Balancer (Martinelli & Mushegian, 2019), Constant Sum Market Maker (CSMM), Hybrid Function Market Makers (HFMM) like Curve Finance (Egorov, 2019), and Dynamic Automated Market Makers (DAMM) such as Bancor (Hertzog et al., 2018). These AMMs, depicted in Fig. 1 (Mohan, 2022), are interconnected and rely on the liquidity pool model.

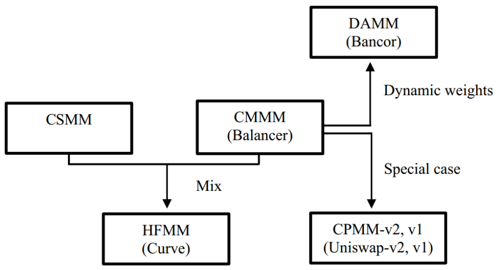

Figure 1. Links between AMMs (Mohan, 2022)

However, both the order book and liquidity pool AMM models have limitations. Liquidity pool initiatives require project initiators to provide initial liquidity, exposing them to withdrawal risks. On the other hand, order book systems require significant token trading volumes to minimize matchmaking durations, posing challenges during periods of low volume. Additionally, there's a risk of market manipulation by vested entities.

As a financial mechanism, the bonding curve concept was introduced by S. De la Rouviere (2017) as a means of facilitating asset trading and price discovery on the blockchain. Unlike traditional markets, bonding curves execute trades automatically based on predetermined mathematical algorithms inspired by market makers. These algorithms, often represented by nonlinear curve functions like exponential or sigmoid functions, dynamically adjust token prices as the token supply changes, enabling adaptive price discovery (Balasanov, 2018).

Despite its potential, the widespread adoption of bonding curves in DeFi has been limited due to a lack of comprehensive theoretical research. Many fundamental questions remain unanswered, particularly regarding its practical application in specific financial scenarios. However, the flexibility and scalability of bonding curves make them an important innovation in the blockchain field, offering new avenues for decentralized finance (DeFi) applications. For example, the model based on bonding curves has been proposed by Titcomb (2019) to enable its application in decentralized governance and incentives.

While bonding curves hold promise as a market-making mechanism, they have yet to dominate the decentralized financial market. Currently, they are only utilized in niche application scenarios due to the limited understanding of their underlying theoretical principles and practical implementation challenges. However, bonding curves offer unique characteristics that distinguish them from traditional order books and liquidity pools (LP). Therefore, this paper aims to delve deeply into the mathematical principles of bonding curves and explore their application in diverse decentralized trading markets. Through meticulous analysis, we endeavor to identify potential application pathways and integration strategies into various financial scenarios, ultimately offering new insights and inspiration for the broader adoption of bonding curves in DeFi and addressing existing limitations of liquidity pool-based and order book-based AMMs.

2. Theoretical Framework

The bonding curve illustrates the correlation between token price and token supply, typically relying on a pegged token to determine pricing, known as an "Anchor Token." Commonly used anchor tokens include ETH, USDT, and other widely accepted tokens. Changes in token supply can result from two factors: alterations in the global token status, achieved through actions like minting or burning, and shifts in the internal liquidity of the bonding curve, accomplished through transfers or redemptions. The choice between these two methods depends on the specific application scenario of the bonding curve. The former is often employed for token launches with adaptive maximum supply, while the latter is typical for token launches with fixed maximum supply and the addition and utilization of unilateral liquidity.

Bonding Curve is expressed as:

$$p = f(s)$$

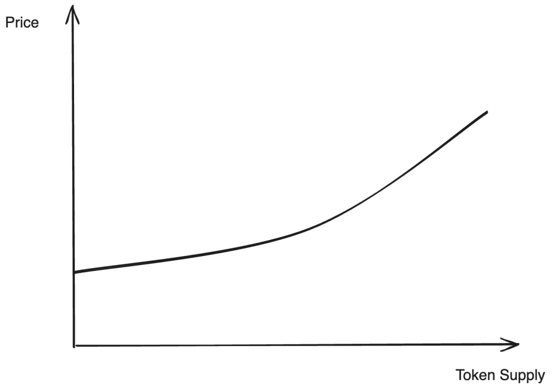

Figure 2. Schematic diagram illustrating a bonding curve using the exponential function as an example

In this equation, p represents the token price, and s represents the token supply. Typically, within the defined domain, $f(s)$ is a non-decreasing integrable function.

The relationship between the circulating supply of tokens in a bonding curve and the number of anchor tokens in the bonding curve is as follows:

$$\frac{ds_a}{ds} = p = f(s)$$

In the context of token launches, deploying a bonding curve doesn't require providing initial asset liquidity but rather increases token circulating supply via methods like Mint or Deposit, and decreases it through methods like Burn or Redeem. With a fixed maximum supply, tokens are all minted into the bonding curve contract during the launch. Users engage with the contract through deposit and redeem actions to complete transactions. When the maximum supply is adaptive, users use mint and burn methods to buy and sell tokens, following similar principles. In a bonding curve, liquidity consists solely of the anchored token, typically locked within the Curve, akin to Total Value Locked (TVL). Since the initial token supply is 0, the TVL expression is:

$$TVL = s_a = \int_0^s f(s)ds$$

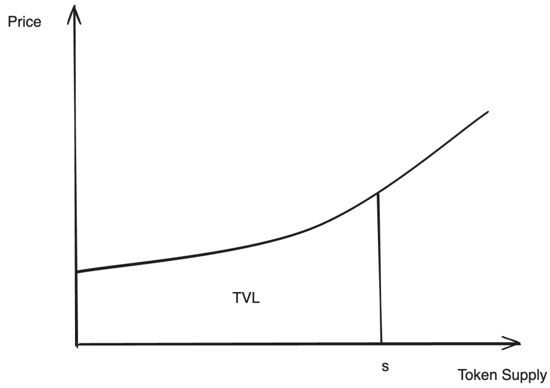

Figure 3. Schematic diagram of TVL in bonding curve

In the scenario of unilateral liquidity, the bonding curve encompasses liquidity for both the trading token and the anchored token of the trading pair. While liquidity for the token needs initialization, the initial liquidity for the anchored asset is 0. When liquidity for the anchored asset is injected externally, it alters the liquidity status of the token asset. Some tokens exit the bonding curve, replaced by the liquidity of the anchor token provided by the liquidity provider. In this scenario, the bonding curve serves as a market maker function akin to a liquidity pool (LP) or an order book. However, unlike traditional setups, initializing liquidity in the bonding curve only requires liquidity for one-sided tokens without involving both tokens. There exists a specific relationship between the change in token liquidity and the change in liquidity of the anchored token:

$$\Delta s_a = \int_s^{s+\Delta s} f(s)ds$$

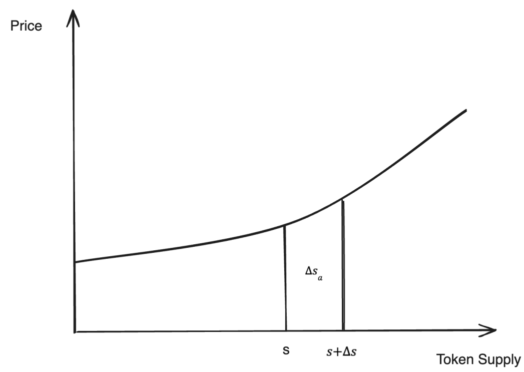

Figure 4. Relationship between the change in token liquidity and the change in liquidity of the anchored token

2.1. Domain of bonding curve

The supply of tokens in bonding curve has a continuous and finite domain (D_1, D_2, D_3, ... , D_n). In various domains, different function expressions may apply. The left endpoint sequence of the domain can be denoted as $(s_1, s_2, s_3, ..., s_n)$, with each domain expressed as $D_i = [s_i, s_{i+1})$. The function expression within each domain is $f_i(s)$. When the supply quantity has a maximum value s_{max}, the last domain range is $[s_n, s_{max})$. Conversely, when the maximum supply is infinite, the last domain range is $[s_n, \infty)$.

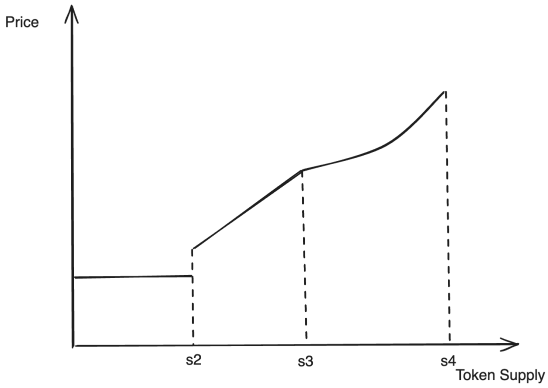

Figure 5. Domain of bonding curve

As bonding curve is an integrable, non-decreasing function, it must satisfy:

$$f_i(s_{i+1}) \leq f_{i+1}(s_{i+1})$$

Additionally, the supply starting point of the bonding curve should begin from 0, indicating:

$$s_1 = 0$$

In the actual calculation process, we first determine the definition domain in which the new supply falls after the state changes, and then compute the specific value within that domain. To simplify the calculation, we can initially compute the maximum liquidity value of the anchored asset in each definition domain:

$$L_i = \int_{s_i}^{s_{i+1}} f_i(s)ds$$

When a user initiates a transaction that changes the token supply, the new state should satisfy the definition domain interval k, which can be expressed as:

$$s_k \leq s + \Delta s < s_{k+1}$$

When a user initiates a transaction that changes the liquidity ΔL of the anchored token, the new state should satisfy the definition domain interval *k*, which can be expressed as:

$$\int_0^{s_k} f(s)ds \leq L + \Delta L < \int_0^{s_{k+1}} f(s)ds$$

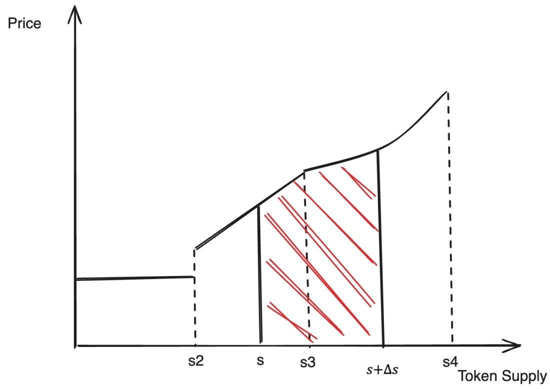

Figure 6. Schematic illustrating the relationship between supply and TVL changes

Piecewise integration is necessary when the initial domain interval is different from the new domain interval.

2.2. Processing of arbitrary functions

To simplify contract calculations, functions within each domain can undergo processing. The function expression in each domain must be continuous and non-decreasing. While exponential functions present an exception, other expressions can be streamlined through Taylor series expansion (Taylor, 1775).

Within the domain interval s_i, the function expression can be converted into a polynomial form:

$$f_i(s) = \sum_{n=0}^{\infty} \frac{f^n(s_i)}{n!}(s - s_i)^n$$

In engineering, certain tools are available to assist users in converting functions.

2.3. Validate errors

To ensure accuracy, we perform error checking by evaluating the Taylor expansion at the right endpoint of each domain. We aim to minimize the error to be less than a predetermined threshold, denoted as σ. This process helps determine the optimal number of terms, denoted as m, for the Taylor series expansion.

$$\left| \sum_{n=0}^{n=m} \frac{f^n(s_i)}{n!}(s_{i+1} - s_i)^n - f_i(s_{i+1}) \right| \leq \sigma f_i(s_{i+1})$$

In order to simplify the calculation, m generally takes the minimum possible value.

2.4. Trading slippage

Slippage, a common occurrence in token trading via bonding curve, is calculated through the following process:

If the current token supply is s_0 and the current token price is p_0, when a user seeks to buy/sell an amount Δs of tokens, the change in the liquidity of the anchor token is given by:

$$\Delta s_a = \int_{s_0}^{s_0+\Delta s} f(s)ds$$

Slippage is:

$$Slippage = \left| \frac{\frac{\Delta s_a}{\Delta s} - f(s_0)}{f(s_0)} \right|$$

Users can set slippage tolerance to ensure that transactions do not experience significant slippage.

2.5. AMM compatibility

All AMM models, including order books, liquidity pools, and bonding curves, fundamentally revolve around an expression linking liquidity and token price. In essence, they share a common foundation. Considering a token A being traded, the relationship between liquidity and price in these models can be as follows:

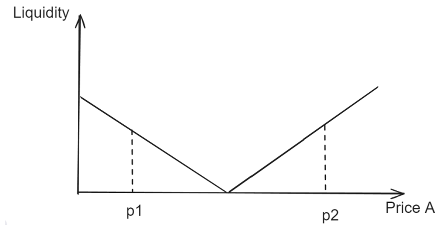

Figure 7. Order book AMM liquidity-price relationship

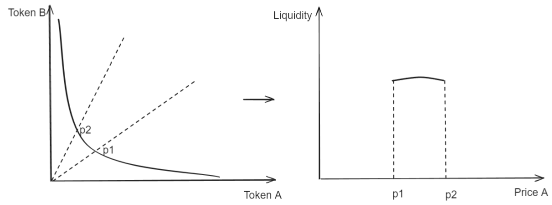

Figure 8. Liquidity pool AMM liquidity-price relationship

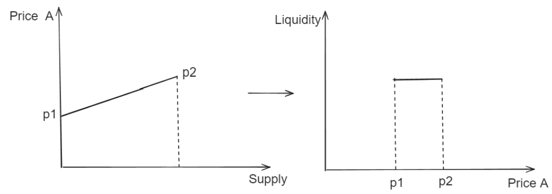

Figure 9. Bonding curve AMM liquidity-price relationship

3 Applications of Bonding Curve Models

In this chapter, we will explore the multifaceted applications of the bonding curve within decentralized finance.

Firstly, the bonding curve offers a means to launch new tokens fairly. By configuring parameters and curve shapes, it enables equitable token launches without centralized control, thereby facilitating decentralized fundraising and token issuance for various projects.

Secondly, with bonding curve liquidities, unilateral liquidity clusters can be established for token pairs, providing unilateral liquidity for new token trading pairs. Liquidity providers can provide token liquidity into the curve, ensuring ample liquidity for trading while retaining the flexibility to set discrete and independent fee ratios programmatically.

Additionally, the bonding curve serves as a decentralized solution for stablecoin trading, enhancing capital utilization. Through the creation of stable and unanchored zones, it enables the abundant supply of tokens in stable exchange rate zones. This mitigates the risk of unilateral liquidity depletion, particularly in cases of anchored token depreciation, thus facilitating stable currency transactions and offering more robust transaction paths for decentralized finance.

Overall, the bonding curve represents an innovative liquidity mechanism and token launch solution with vast potential applications. Its significance in the digital currency realm is profound, promising to reshape the landscape of decentralized finance.

3.1 Token launch

Fair launching represents a unique application scenario where the bonding curve outshines other market makers like order books and liquidity pools. It facilitates the early and equitable launch of tokens without necessitating the issuer to provide initial liquidity of anchored tokens for pricing.

In a fair launching setup, token supply adjusts concurrently with the pricing process. Typically, token supply is regulated solely through the bonding curve contract, with no external token supply involved.

The key advantages of fair launching are twofold. Firstly, it ensures the fairness of token issuance by mandating that all tokens are issued through the bonding curve rather than being generated arbitrarily. Secondly, tokens launched fairly exhibit stable value and flexible supply dynamics. Market demand automatically balances token price and supply, with each token backed by a 1:1 ratio of anchor tokens, thus providing robust value support.

3.1.1 Tax module

In a fair minting scenario, token supply cannot be created arbitrarily, and token issuers cannot generate income by directly selling their token shares in the traditional manner. However, with a bonding curve, token issuers can generate continuous income through the implementation of a tax module. This revenue model is fairer to participants because it allows token issuers to earn higher revenue as the token's trading volume increases.

Typically, token owners have the option to establish two-way taxes: token minting (purchase) tax (t_m) and token burning (sale) tax (t_b). These taxes can either be constant or tied to specific conditions within the bonding curve, such as current supply, anchored token liquidity, or historical accumulated tax. The taxes are applied to the anchored token.

When utilizing the tax module, the tax calculation process unfolds as follows:

When a user intends to purchase a quantity Δs of tokens, the required minting tax to be paid is:

$$T_m = t_m \int_{s_0}^{s_0+\Delta s} f(s)ds$$

The total number of anchor tokens that users need to pay is calculated as follows:

$$s_a = (1 + t_m) \int_{s_0}^{s_0+\Delta s} f(s)ds$$

When the user wants to buy(mint) Δs_a amount of tokens, the tax paid is determined by:

$$T_m = t_m \Delta s_a$$

When a user wants to sell Δs amount of tokens, the tax paid is calculated as:

$$T_m = t_m \int_{s_0-\Delta s}^{s_0} f(s)ds$$

he number of anchor tokens that users can redeem is:

$$s_a = (1 - t_m) \int_{s_0-\Delta s}^{s_0} f(s)ds$$

Additionally, the tax module can implement varying taxes across different domains. For instance, the minting tax could be set to 100% in the early definition domain to fulfill fundraising requirements, meaning all anchor tokens paid for minting are transferred to the token issuer.

3.1.2 Pre-mint module

The pre-mint module is typically executed concurrently with the deployment of the bonding curve contract to ensure it becomes the initial transaction on the Curve. This functionality effectively mitigates front-running by blockchain bots.

When the token issuer employs the pre-mint module, the anchored tokens must be deposited into the bonding curve contract during its deployment. If the pre-mint amount is s_p, the number of anchor tokens required to be transferred during contract deployment is:

$$s_a = \int_0^{s_p} f(s)ds$$

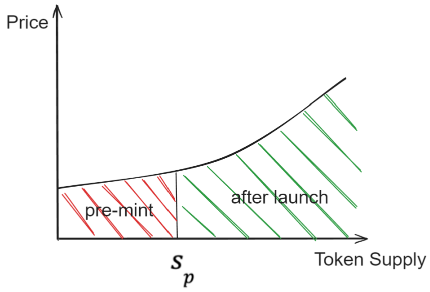

Figure 10. Token pre-mint module

The pre-mint function enables the project issuer to achieve the lowest theoretical token price. If token pre-sales were conducted before deploying the bonding curve, those tokens can also be redeemed using the pre-minted ones.

Furthermore, when the bonding curve of a token is increasing, the token price after pre-minting will create a ratio to the average pre-minting price. When the amount of minted tokens is s_p, the ratio q is:

$$q = \frac{f(s_p)s_p}{\int_0^{s_p} f(s)ds}$$

For any non-decreasing function:

$$q \geq 1$$

This implies that after a token issuer conducts early fundraising with a certain initial market value, if the token price corresponding to this value is used as the initial trading price, the token issuer only needs $\frac{1}{q}$ worth of anchor tokens for pre-sale.

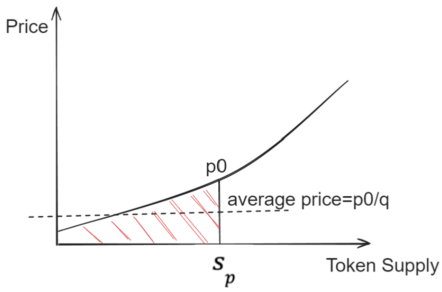

Figure 11. Pre-mint in case of pre-sale

3.1.3 Token unlocking module

In a token launch scenario, early buyers often enjoy significant price advantages. To balance the interests of early and subsequent users and mitigate front-running transactions, an unlocking module can be implemented.

This module directs minted tokens to an unlocking contract rather than directly to users, releasing them based on predetermined conditions.

Typically, the unlocking period is determined by the token supply at minting, with smaller supplies corresponding to longer unlocking periods.

For instance, tokens minted in the $[0, s_1)$ supply range might have a 12-month unlocking period, while those in $[s_1, s_2)$ might unlock in 6 months.

Alternatively, unlocking periods can be set continuously, such as $[d_1, d_2)$ within the $[0, s_1)$ interval. The relationship between unlocking cycles and supply is thereby established:

$$d(s) = d_1 + \frac{d_2 - d_1}{s_1} s$$

Then, if a user mints a number of tokens Δs within this range, the number of tokens R_d they can obtain each day is:

$$R_d = \int_s^{s+\Delta s} \frac{1}{d_1 + \frac{d_2 - d_1}{s_1} s} ds$$

Additionally, the unlocking module can be combined with the tax module and pre-minting module to execute more intricate economic models.

3.1.4 Other modules

In addition to the mentioned modules, additional ones can be established in a modular manner, such as:

- Delayed trading module: imposing trade restrictions until a specified block height post contract deployment.
- Dynamic tax module: adjusting tax rates dynamically based on factors like real-time Total Value Locked (TVL) size.
- Automatic buyback or burning module: automatically repurchasing tokens or initiating token burns using collected taxes.
- Airdrop module: distributing pre-minted tokens through airdrops to community members.
- Lending and borrowing module: unlocking anchored token liquidity within the bonding curve, featuring self-liquidation without third-party intervention.

3.2 Unilateral liquidity addition and management

In addition to enabling fair minting of newly launched tokens, bonding curves can also build unilateral liquidity clusters for existing tokens. The purpose of this implementation is similar to the order book model of centralized exchanges and the automatic market maker model of Uniswap's liquidity pool.

The establishment of the bonding curve unilateral liquidity cluster requires the provision of initial liquidity for one-sided tokens. After providing the initial liquidity of the token, the bonding curve will form a trading pair of the token with the pegged token.

3.2.1 Establishment of initial liquidity

When constructing bonding curve's liquidity, three parameters must be established: the initial liquidity of the token (s_{max}), the initial token price (p_{min}), and the final token price (p_{max}).

Additionally, defining the curve of the token within the liquidity range is essential. In the unilateral liquidity cluster scenario, the curve's domain is $[0, s_{max}]$, with s_{max} representing the token's initial liquidity. The curve expression is denoted as $p = f(s)$, satisfying the conditions $p_{min} = f(0)$ and $p_{max} = f(s_0)$.

For instance, let's consider building a liquidity cluster for ETH using the bonding curve: Initially, 1,000 ETH is deposited into the bonding curve contract as the initial liquidity. The anchor token is set to USDT, with an initial price of 2,000 USDT/ETH and a final price of 4,000 USDT/ETH. Linear functions can be selected as the curve. Thus, the expression for the ETH-USDT liquidity bonding curve would be:

$$p = 2000 + 2s$$

The domain of s is [0,1000].

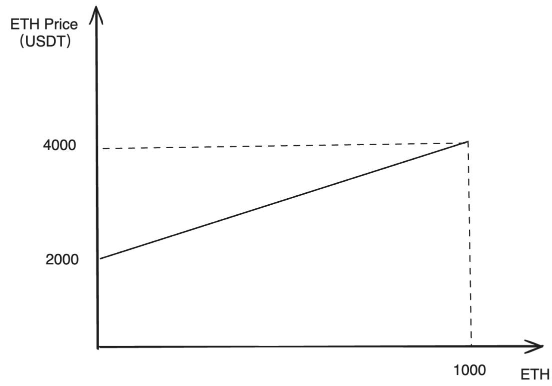

Figure 12. Liquidity distribution example of ETH

3.2.2 Trading on the unilateral liquidity cluster

After the unilateral liquidity cluster is established, users can engage in transactions between trading tokens and anchored tokens within it. Initially, the first transaction can only acquire tokens using the anchored token, achieved through the redeem() method. This action involves obtaining tokens from the initially added liquidity, rather than directly minting new tokens via the token contract.

To obtain s tokens from the unilateral liquidity cluster, the required number of anchor tokens to be paid is calculated as follows:

$$\Delta s_a = \int_{s}^{s+\Delta s} f(s)ds$$

The conditions to be met are:

$$s + \Delta s \leq s_{max}$$

As the bonding curve transitions from its initial state, the unilateral liquidity cluster contains both tokens and anchored tokens. Users can then exchange their tokens for the anchored token.

To exchange Δs tokens for anchored tokens, the number of anchored tokens obtained is:

$$\Delta s_a = \int_{s-\Delta s}^{s} f(s)ds$$

The conditions to be met are:

$$s - \Delta s \geq 0$$

3.2.3 Liquidity distribution

Once the bonding curve moves beyond its initial state, liquidity is available for both the token and the anchored token. If the bonding curve is represented as $p = f(s)$, with the definition domain $[0, s_{max}]$, then the quantities of liquidity for the token and the anchored token are $s_{max} - s$ and $\int_0^s f(s)$, respectively.

Similar to Uniswap V3, liquidity on the bonding curve is distributed across price ranges. If the current state is (s_0, p_0), then in the range where $p < p_0$, only the liquidity of the anchor token exists, whereas in the range where $p > p_0$, only the liquidity of the token exists.

Let $s = g(p)$ be the inverse function of $p = f(s)$.

In the interval where $p < p_0$, the liquidity distribution of the anchored token is denoted as $l(p)$, satisfying:

$$l(p)dp = pds$$

thus,

$$ds = g'(p)dp$$

hence,

$$l(p) = pg'(p)$$

In the interval where $p > p_0$, the liquidity distribution of the anchored token is denoted as $l(p)$, and it satisfies:

$$l(p)dp = ds$$

Then

$$l(p) = g'(p)$$

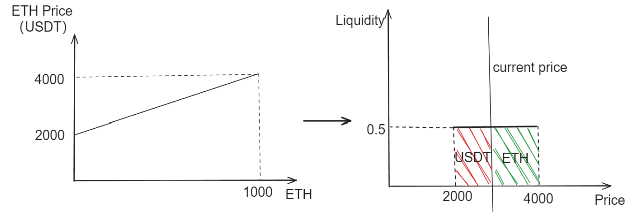

Figure 13. Liquidity distribution example of ETH

3.2.4 Liquidity aggregation

In the bonding curve's unilateral liquidity cluster setup, each bonding curve unilateral liquidity operates independently, allowing users to customize curve parameters and fee modules. However, liquidity aggregation across multiple pools is possible through Hooks. These Hooks enable an aggregator to optimize trading strategies and execute transactions across various bonding curve pools.

The process of liquidity aggregation simplifies into a convex optimization problem, falling within the domain of quadratic positive definite problems, ensuring a solution within unit time—a *P* problem. However, at the AMM's base layer in the liquidity pools, this task becomes a *NP* problem, meaning a solution exists but cannot be computed within unit time.

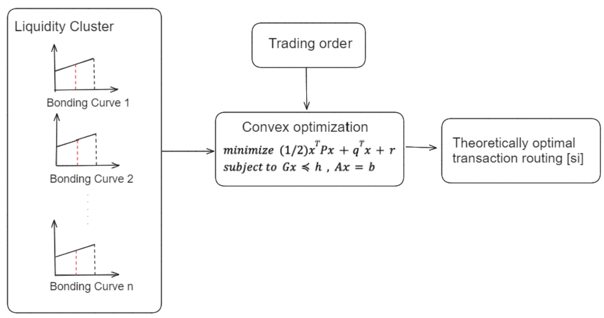

Figure 14. Unilateral liquidity aggregation

3.2.5 Trading slippage

In a user transaction scenario, slippage occurs when purchasing or selling Δs amount of tokens using the anchor token. Assuming the bonding curve's current state is (s_0, p_0), the transaction's slippage is calculated as follows:

$$Slippage = \left| \frac{\frac{\Delta s_a}{\Delta s} - f(s_0)}{f(s_0)} \right|$$

and

$$\Delta s_a = \int_{s_0}^{s_0+\Delta s} f(s)ds$$

3.2.6 Feature analysis

In the context of liquidity pools, users can only initially provide liquidity for one token, while the other token is treated as the anchor token.

Adjusting the bonding curve's functional expression allows for different distributions of token liquidity. For instance, employing a linear function results in a liquidity distribution akin to that of Uniswap V3, albeit without the Tick Space concept (Adams et al., 2021), opting instead for continuous liquidity distribution.

Bonding Curve's liquidity providers benefit from a more adaptable strategy for adding liquidity, enabling a diversified income approach. For instance, users can set up a bonding curve within a specific price range, such as 2700 USDT to 2800 USDT for ETH, and impose a 5% transaction fee. This setup allows users to profit from price fluctuations between 2500 USDT and 3000 USDT, with each price crossing yielding a 5% profit.

In summary, bonding curve's unilateral liquidity cluster application offers enhanced flexibility and potential profitability for liquidity providers in decentralized finance ecosystems.

3.3 Stablecoin trading pool

Transactions involving stablecoins constitute a significant segment within the digital currency trading realm. Curve, integrating constant sums and products, has pioneered the stablecoin market maker model (Egorov, 2019). Nevertheless, this model presents certain drawbacks:

- Adjusting the A value in response to liquidity changes is necessary to maintain curve integrity, yet this parameter's abstract definition poses comprehension challenges for users.
- Liquidity depletion risks arise when tokens are de-anchored beyond a certain threshold, leading to diminished liquidity for specific tokens.
- Suboptimal liquidity utilization persists due to users' limited tolerance for exchange rate deviations, resulting in narrow transaction liquidity ranges.
- Balancing liquidity on both sides is essential, but achieving this equilibrium with a single-coin initialized pool is challenging.

These challenges stem from Curve's curve's lack of flexibility and limited adjustability.

Curve's market maker curve:

$$\chi(x + y) + xy = \chi D + (\frac{D}{n})^n$$

in

$$\chi = \frac{Axy}{(\frac{D}{n})^n}$$

To tackle these challenges, integrating twin liquidity bonding curves (e.g., USDC-USDT and USDT-USDC bonding curves) into bonding curve's aggregation presents a more versatile stablecoin market maker model, which is proposed by Burve Labs.

3.3.1 Stablecoin bonding curve

The bonding curve model for stablecoins can be delineated into two segments using a piecewise function:

1. **Stable Zone:** Characterized by a constant exchange rate, denoted as $p = C$.

- Definition Domain:$0 \leq s \leq s_u$
- s_u represents the token supply at the initiation of unanchoring.

2. **De-Mooring Zone:** Where the exchange rate deviates from the stable value.

- In this zone, the expression of the bonding curve is $p = C\frac{s_{max} - s_u}{s_{max} - s}$
- Definition Domain:$s_u \leq s \leq s_{max}$
- Here, s_{max} signifies the maximum supply of the token.

In the stability zone, the exchange rate remains constant, ensuring predictability. However, as the token supply surpasses s_u and enters the de-mooring zone, the exchange rate becomes dynamic, adjusting based on the token supply. The graphical representation illustrates this transition.

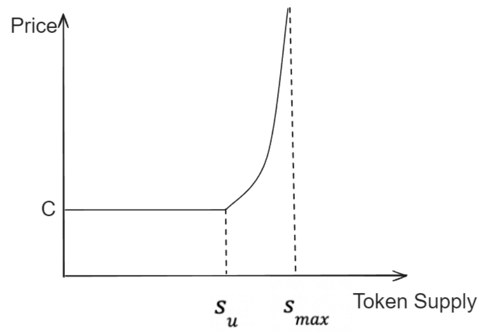

Figure 15. Piecewise function indication

It's noteworthy that integral of $\int_{s_u}^{s_{max}} C\frac{s_{max}-s_u}{s_{max}-s}$ ds within the de-mooring zone yields an infinite value, suggesting no unilateral liquidity shortage for the token.

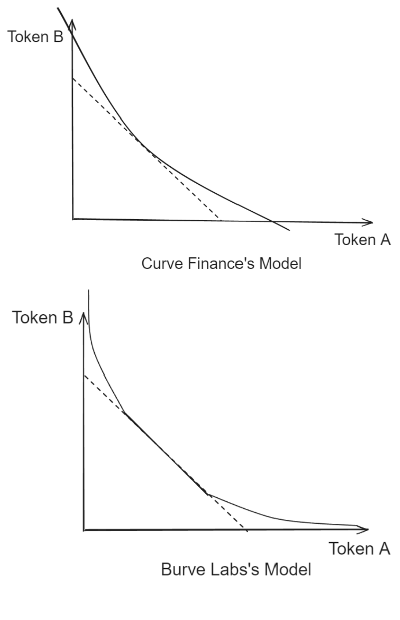

Figure 16. Comparison of Curve and Burve's stablecoin liquidity market-making curves

3.3.2 Creation of stablecoin liquidity pair

When adding liquidity for two stablecoins A and B, we simultaneously create two bonding curves. One curve uses Coin A as the anchor token, while the other utilizes Coin B as the anchor token.

For the curve anchored by Coin A, an initial liquidity of a specific amount of Coin B tokens is necessary. Conversely, the curve anchored by Coin B requires an initial liquidity injection of a certain amount of Coin A tokens.

Additionally, we establish the unanchored domain of the token, typically represented as a percentage, denoted as γ.

However, it's crucial to note that:

$$s_u = \gamma s_{max}$$

The expressions for these two bonding curve curves are as follows:

For the curve anchored by Coin A:

$$p^A = \begin{cases} C^A & (0 \leq s^A \leq s_u^A) \\ C^A \frac{s_{max}-s_u^A}{s_{max}-s^A} & (s_u^A < s^A \leq s_{max}^A) \end{cases}$$

And for the curve anchored by Coin B:

$$p^B = \begin{cases} C^B & (0 \leq s^B \leq s_u^B) \\ C^B \frac{s_{max}-s_u^B}{s_{max}-s^B} & (s_u^B < s^B \leq s_{max}^B) \end{cases}$$

in

$$C^B = \frac{1}{C^A}$$

3.3.3 Stablecoin trading

When exchanging Δs amount of stablecoin A for stablecoin B, there are two potential paths: acquiring $\int_{s_0-\Delta s}^{s} f^A(s^A)ds^A$ worth of B tokens on the $p^A = f^A(s^A)$ curve, or obtaining B tokens on the $p^B = f^B(s^B)$ curve.

This creates multiple exchange paths for the transaction. To automatically select the optimal exchange path, we utilize Hooks, a decentralized transaction path selector akin to Uniswap's routing. By implementing route optimization algorithms, Hooks help users find the most efficient transaction path, thereby reducing transaction costs.

In the initial state, each curve holds liquidity for only one of the tokens. As liquidity changes, Hook prioritizes the bonding curve corresponding to the token with decreased liquidity.

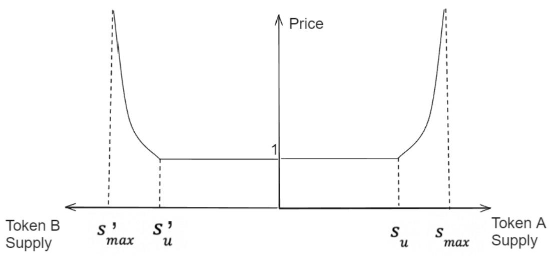

Figure 17. Aggregated curve when C=1

For instance, consider an initial liquidity of 1M tokens for both A and B, with an exchange rate of 1. If, in a subsequent state, the liquidity of tokens A and B becomes 0.8M and 1.2M respectively, Hook will prioritize the bonding curve with $p^A = f^A(s^A)$.

Moreover, if a trade cannot be completed in a single transaction due to insufficient liquidity on a particular curve, it will be split into two trades. The first transaction restores both curves to their initial state, while the second transaction is completed on the alternative curve.

Discussion

The theoretical framework outlined in previous chapters has been systematically summarized and extensively discussed. Building upon this foundation, the corresponding functions have been implemented and verified. The smart contract for the token launch component can be accessed at https://github.com/BurveProtocol. The addition and invocation of unilateral liquidity, as well as the stablecoin trading aspect, will be introduced in contract versions 2.0 and 3.0 respectively.

The outcomes of this research address numerous pain points within the current crypto ecosystem and cater to the token launch requirements of a wide array of projects.

Conclusion

This article explores the application of the bonding curve, an automated market maker model, across various decentralized trading scenarios, including fair launching, unilateral liquidity cluster and stablecoin trading.

Compared to other market maker models, the bonding curve introduces novel features and unique advantages to these application scenarios.

Fair Launching: Bonding Curve facilitates fair token minting, enabling supply adaptation and price discovery without requiring initial liquidity for curve creation. The use of piecewise curves and dynamic tax modules allows for more flexible economic model design, effectively balancing the rights and interests of token issuers and holders.

Unilateral Liquidity Cluster: Bonding Curve implements an automatic market maker mechanism between any tokens, akin to Uniswap LP and order books. By adjusting curve parameters, liquidity allocation can be more flexible across different price ranges. Each token's liquidity functions independently as a bonding curve, and liquidity across the network aggregates through Hook. Providers can set different parameters on each bonding curve for a more flexible liquidity income strategy.

Stablecoin Trading: Bonding Curve facilitates more efficient stablecoin trading, enabling users to intuitively set stablecoin de-anchoring parameters, improving liquidity utilization. Twin liquidity settings can be implemented for greater transaction flexibility and to mitigate unilateral liquidity shortages.

In conclusion, the bonding curve emerges as an extremely flexible and scalable automated market maker mechanism. It injects vitality into decentralized transactions and finance, inspiring innovative applications. Its introduction signifies not only technological advancement but also a pivotal shift in industry development. Through the bonding curve, we anticipate a more open and inclusive financial system, providing barrier-free, efficient, and secure financial services, thereby propelling the vigorous growth of the entire decentralized financial ecosystem.

References

Adams, H., Zinsmeister, N., & Robinson, D. (2020). Uniswap v2 Core. Retrieved from https://uniswap.org/whitepaper.pdf

Adams, H., Zinsmeister, N., Robinson, D., Salem, M., & Keefer, R. (2021). Uniswap v3 Whitepaper. Retrieved from https://uniswap.org/whitepaper-v3.pdf

Balasanov, S. (2018, December). Bonding curves in depth: Intuition & parametrization. Relevant Community. Retrieved from https://medium.com/relevant-community/bonding-curves-in-depth-intuition-parametrization-d3905a681e0a

De La Rouviere, S. (2017). A bonded curation community. Retrieved from https://medium.com/@simondlr/a-bonded-curation-community-247f14a6de04

De La Rouviere, S. (2017). Tokens 2.0: Curved token bonding in curation markets. Retrieved from https://medium.com/@simondlr/tokens-2-0-curved-token-bonding-in-curation-markets-1764a2e0bee5

Egorov, M. (2019). StableSwap – Efficient Mechanism for Stablecoin Liquidity. Retrieved from https://www.curve.fi/stableswap-paper.pdf

Hertzog, E., Benartzi, G., & Benartzi, G. (2018). Bancor Protocol. Retrieved from https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf

Martinelli, F., & Mushegian, N. (2019). Balancer: A Non-custodial Portfolio Manager, Liquidity Provider, and Price Sensor. Retrieved from https://balancer.finance/whitepaper/

Mohan, V. (2022). Automated market makers and decentralized exchanges: a DeFi primer. Financ Innov, 8, 20. https://doi.org/10.1186/s40854-021-00314-5

Pourpouneh, M., Nielsen, K., & Ross, O. (2020). Automated Market Makers (IFRO Working Paper No. 2020/08). University of Copenhagen, Department of Food and Resource Economics (IFRO). Retrieved from https://hdl.handle.net/10419/222424

Schär, F. (2021). Decentralized finance: On blockchain and smart contract-based financial markets. Federal Reserve Bank of St. Louis Review, 103, 153–174. https://doi.org/10.20955/r.103.153-74

Taylor, B. (1715). Methodus incrementorum directa et inversa (pp. 329–332).

Titcomb, A. (2019, April). Deep dive: Augmented bonding curves. Giveth. Retrieved from https://medium.com/giveth/deep-dive-augmented-bonding-curves-3f1f7c1fa751

Voshmgir, S. (2019). Token Economy: How Blockchains and Smart Contracts Revolutionize the Economy (Vol. 1). BlockchainHub.

Xu, J., Vavryk, N., Paruch, K., & Cousaert, S. (2021). SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols. Retrieved from https://arxiv.org/abs/2103.12732