DeltaFi V2 – A Next-Gen Decentralized Exchange

Fair Prices, Sustainable Yields, and Capital Efficiency

DeltaFi Labs

April 2022

Abstract

DeltaFi is an innovative decentralized exchange (DEX) powered by the most efficient automated market maker (AMM) on Solana, termed the Intelligent Market Maker (IMM). DeltaFi proposes an innovative bond curve on top of real-time, on-chain oracle pricing to eliminate liquidity loss and achieve sustainable yields for liquidity providers (LP). The predictive bond curve applies to both volatile assets and stable swaps. Furthermore, native cross-chain swaps are enabled through Wormhole.

Disclaimer

This whitepaper outlines the current and future protocol design of DeltaFi. It is made available solely to provide information and is not a statement of future intent. The information contained in this whitepaper has not been approved by any regulatory authority and may be amended, updated, or otherwise modified.

This whitepaper is not a prospectus or offer document and is not intended to constitute an offer of securities or a solicitation for investment in securities in any jurisdiction. The content contained in this document should not be interpreted as professional, financial, tax, legal, or business advice, nor as a provision for any advice.

Table of Contents

Executive Summary

DeltaFi has revolutionized the DEX landscape with the development of a high-performance AMM, dubbed an Intelligent Market Maker. Many mainstream DEXs that rely on the constant product formula encounter high impermanent loss, inefficient pricing, and low capital efficiency. That's not us. DeltaFi proposes a predictive bond curve. The predictive bond algorithm maintains a fixed ratio between base and quote tokens in the pool by incentivizing users to swap in a desirable direction to keep the pool in a balanced state. Our design protects liquidity providers from losing value as they trade, deposit, and withdraw their assets on the exchange.

We pride ourselves on efficiency, which is why we built DeltaFi on Solana, the fastest blockchain in the world. We believe that Solana will become the liquidity hub in a multi-blockchain world, and have built a cross-chain exchange on Solana with the Wormhole to provide flexibility to our users. Moreover, we've adopted the Pyth Network to provide real-time slot by slot on-blockchain market price information, ensuring the most up-to-date price information. Coupled with the predictive bond algorithm, users will longer have to worry about losing the value of their assets when providing liquidity or withdrawing tokens from a pool.

Decentralized Finance (DeFi) has taken the global financial services industry by storm. DeFi has many applications, including lending, borrowing, and trading - promising to cut out the middleman and make financial services trustless, more transparent, and accessible to all. As more people become disillusioned by traditional financial institutions, they turn to DeFi, and, more specifically, decentralized exchanges. DEXs offer more control and flexibility without relying on intermediaries to dictate rules imposed by centralized exchanges.

The breakthrough innovations occurring on cryptocurrency and blockchain-based platforms are only outpaced by the growing interest in DeFi. This whitepaper aims to outline the design of DeltaFi, and illustrate an innovative solution to the common problems faced by other DEXs that utilize the traditional automated market maker design. Altogether, we've created a layer 2 solution that advances DEX technology and puts users first. We're so excited to share what we have built for you.

1 Introduction

1.1 Existing solutions

Existing decentralized exchanges on Solana can be classified as orderbook solutions or automated market makers. Orderbook solutions promote an efficient market with an aggregated list of buy and sell orders at the sacrifice of decentralization. On the other hand, AMM solutions democratize market transparency by allowing anyone to deposit liquidity for yield farming and enable trading. AMMs aggregate liquidity for both sides of a trading pair and determine a fair market price with a price formula based on a pool's current liquidity.

Most AMMs adopt the constant product formula. However, it suffers from:

a) High Impermanent Loss: It incentivizes users to buy appreciating tokens and sell depreciating tokens. This can cause significant loss for liquidity providers.
b) Inefficient Pricing: It relies on arbitrageurs to lower price gaps with other exchanges.
c) Low Capital Efficiency: It uses only a small part of the liquidity in the pool. Some AMMs use concentrated liquidity to resolve at the sacrifice of higher impermanent loss.

1.2 Predictive Bond Curve

In this paper, DeltaFi proposes a predictive bond curve to achieve more efficient pricing and solve the impermanent loss problem. The algorithm maintains a fixed ratio between base and quote tokens in the pool by incentivizing users to swap in a desirable direction to keep the pool in a balanced state. When we eliminate the ratio change, so that we can eliminate impermanent loss. This incentive is immaterial to the price of the token price so that it will not cause any impermanent loss.

Our design protects liquidity providers from losing value as they trade, deposit, and withdraw their assets on the exchange.

The predictive bond curve is detailed further with demonstrated back-testing in this paper.

2 Terms and Notations

2.1 Terminology

Base and Quote Token. It refers to the tokens at each side of the pool. Base and quote tokens are symmetric in terms of swap logic. Therefore, in the technical sections about swap logic, we only focus on trader selling base and buying quote since the opposite operation follows the same reasoning due to the symmetry.

Implied Price. It refers to the price the user gets from trade.

2.2 Notations

Notation	Definition
A	Balanced point of the base token reserve
B	Balanced point of the quote token reserve
A_{norm}	Normalized base token reserve
B_{norm}	Normalized quote token reserve
A_{init}	Base token reserve at the initial state
B_{init}	Quote toke reserve at the initial state
A_0	Target reserve of base token recorded after last deposit/withdraw
B_0	Target reserve of quote token recorded after last deposit/withdraw
A_1	Target reserve of base token calculated for withdrawal
B_1	Target reserve of quote token calculated for withdrawal
a	Current reserve amount of base token
b	Current reserve amount of quote token
T	Minimum ratio of the normalized reserve left
Δa	The current base reserve deviation from the target base reserve, i.e.$B - b$
Δb	The current quote reserve deviation from the target quote reserve, i.e.$A - a$
P	Market price, amount of quote to buy with 1 base
s	Slope of price curve
$g_0(a, b, m)$	With A, B as constant, a, b, m as variable, the amount of quote token traded from m base token
$g(m)$	With A, B, a, b as constant, amount of quote token traded from m base token
c	Confidence interval provided by Pyth Network
$c_{adjusted}$	Adjusted confidence interval with time weighted average price information
o	Oracle price provided by Pyth Network
d	The absolute difference between oracle price and TWAP price
o_{twap}	The TWAP oracle price provided by Pyth Network
r	The volatility ratio is defined as $\frac{d}{c}$
TVL	Total value locked

Table 1. Symbol Notations.

3 Oracle Price Correction

3.1 Oracle Data Provider

DeltaFi adopts Pyth Network, which provides real-time, slot-by-slot on-chain market price information. The Pyth Network is a decentralized, cross-chain financial oracle that provides verifiable data from high-quality nodes. The oracle concatenates price information from multiple trusted market data sources and pushes the newest update to the Solana Network every ~400 milliseconds. DeltaFi takes advantage of oracle price to set a base fair market price for the bond curve.

In addition to the accurate and up-to-date price, DeltaFi leverages additional signals such as time-weighted average price (TWAP) and a confidence interval for spread predictions and risk mitigation. In reality, oracle prices can potentially be delayed, contain noise, and are not ideal for pricing. Nevertheless, such signals are significant inputs for oracle price corrections.

3.2 Confidence Interval Correction

Confidence interval implies a spread for the oracle price. In addition to the confidence interval, TWAP price in the Pyth data history is also a factor to consider price volatility, reflecting the price change trend and market risks.

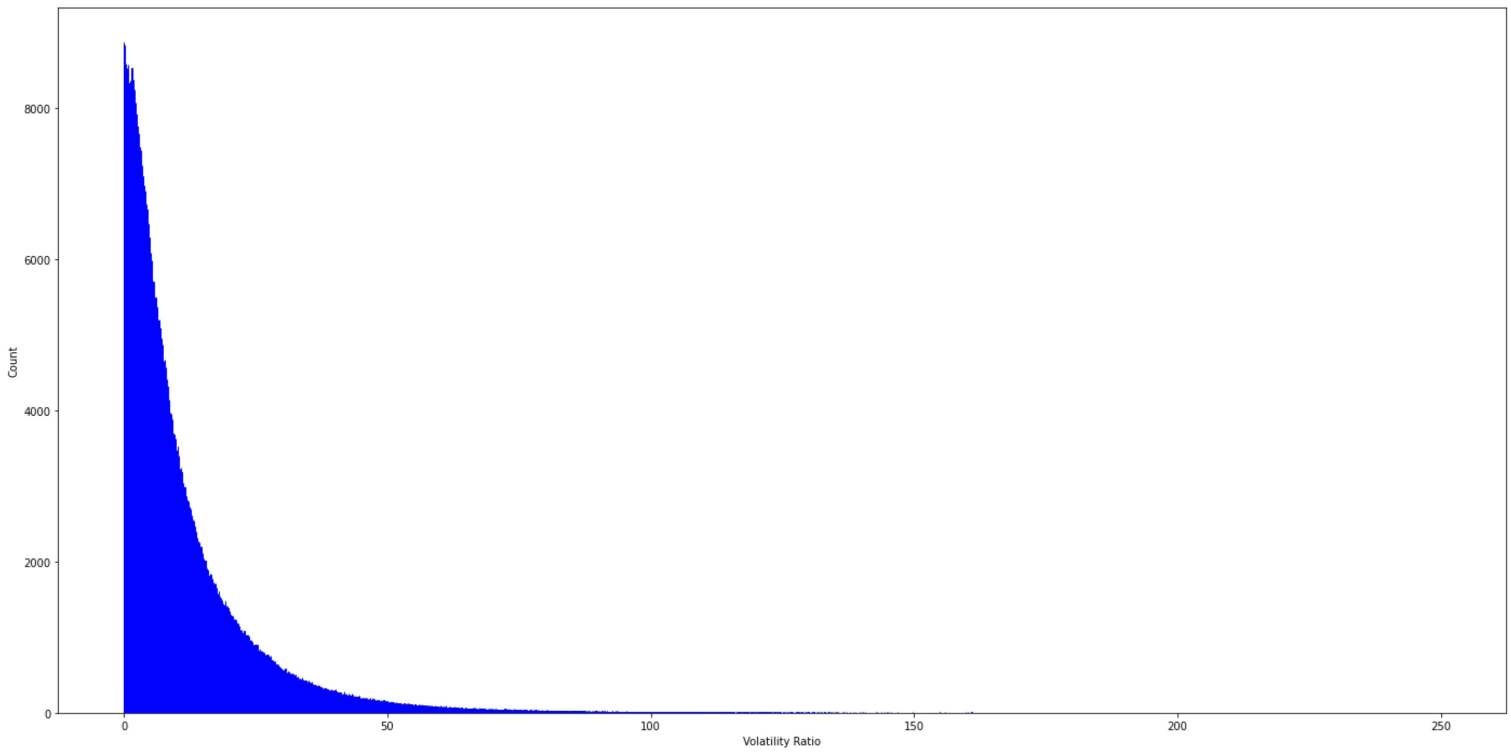

Figure 1. Volatility ratio histogram distribution. It shows slot by slot count of volatility ratio values.

Recall from the notation table, volatility ratio is defined as:

$$r = \frac{d}{c}$$

where d is the absolute difference between oracle price and TWAP price o_{twap}, and c is the confidence interval.

The volatility ratio is used to capture volatile market situations not reflected by confidence interval. For example, if TWAP to oracle price difference is large, the price changes in one direction very quickly. Under such situations, the confidence interval is adjusted with such TWAP information to reflect market volatility. From the volatility ratio distribution, threshold 100 is selected to reflect extreme market volatilities.

if volatility ratio $r > 100$

$$c_{adjusted} = \frac{d}{100}$$

else:

$$c_{adjusted} = c$$

if the order direction is selling base:

$$P = P - c_{adjusted}$$

else:

$$P = P + c_{adjusted}$$

4 Bond Curve Formulation

4.1 Predictive Bond Curve

Because the base and quote token are symmetrical, we only discuss the swap that sells base tokens for quote tokens below.

To start with, the notations are:

- In the pricing formula, we use A, B as balanced reserves
- The current base reserve is a, and the quote reserve is b.
- The current base to quote market price is P (i.e., corrected oracle price), meaning that with market price, we can buy P quote tokens with 1 base token.

The main goal of the pricing method is to encourage traders to swap to get the pool into a balanced state. The balanced state means the pool's base/quote ratio is the same as the initial ratio, $\frac{A}{B}$.

With initial reserve A and B, market price P, with *any* base and quote reserve a and b, we let the price at this certain point is:

$$p(a,b) = P \cdot \frac{A}{B} \cdot \frac{b \cdot s + B \cdot (1-s)}{a \cdot s + A \cdot (1-s)}$$

This guarantees that

1. With $\frac{a}{b} = \frac{A}{B}$, the price is exactly at the market price; the marginal gain is 0 for both swap sides.

2. For the same b, the larger the a is, the more quotes can be exchanged with 1 base token, there will be more advantage of selling base, same for the opposite case.
3. Variables are the slope level of the price curve.

For certain base and quote reserve a, b, with Fixed A, B, P let $g(m)$ be the amount of quote token to get from selling m base token. We let $g(m)$ be the integration result of the price curve from(a, b) to $(a+m, b-g(m))$, so that we can guarantee that:

for any m' < a, any positive integer n,

If $\sum_{i=1}^{n} m_i = m'$

Then $\sum_{i=1}^{n} g_i(m_i) = g(m')$ (Note: $g_i(m)$ represent the curve function with base and reserve position after i-1 small transactions)

With the definitions above:

$$g(m) = \frac{P \cdot A}{B} \cdot \int_0^m \frac{s \cdot (b - g(x)) + (1-s) \cdot B}{s \cdot (a+x) + (1-s) \cdot A} \, dx$$

Solving the differential equation [see addendum], we have:

$$g(m) = \left(b + \frac{1-s}{s} \cdot B\right) \cdot \left(1 - \left(\frac{s \cdot a + (1-s) \cdot A}{s \cdot (a+m) + (1-s) \cdot A}\right)^{\frac{P \cdot A}{B}}\right)$$

Theorem 1. With this price curve, the optimal swap is always the swap that draws the pool back to the target ratio.

Proof. [See addendum]

Theorem 2. Selling and buying one token multiple times is equivalent to selling or buying the total amount in one swap.

Proof. [See addendum]

Theorem 1 guarantees the swap incentive is pointed to the balanced point is at $\frac{a}{b} = \frac{A}{B}$. Theorem 2 ensures that there is no way for a trader to profit from buy/sell combinations, and there is no incentive for the trader to split a large transaction into continuous smaller ones to get a better price.

The above discussions are based on the assumption that P, A, B are constant values, current reserves a, b are moving within a determined curve. However, in a practical usage of such a price curve formula, the market price changes, making the balanced point in the context changes. And because we will collect fees in the real application, the balanced reserve point shifts after a trade.

Therefore, for each trade, we need to calculate the balanced point from the current reserve a, b and target reserves ratio $\frac{A_0}{B_0}$, suppose $A_0 \; and \; B_0$ are initial ratio and we will keep the balanced reserves ratio same as this value all the time.

The process of finding the balanced point will be further elaborated in the addendum.

Rebalancing Incentives. First, bond curve g(m) is visualized to see the impact of pool states to pricing. When the pool is not balanced, the pricing curve gives incentives for arbitrageurs to bring the pool back into balance. Once the pool is balanced, the bond curve price is the same as the market price. Figure 3 shows that:

- When the base reserve is smaller than the target base reserve, the price is more favorable for a trade to sell the base tokens for quote tokens to rebalance the pool. This is illustrated with $a = 90$, and $b = 123.46$ in Figure 2.
- When the quote reserve is smaller than the target reserve, the price is more favorable for a trade to sell the quote tokens for base tokens to rebalance the pool. The curve is symmetric. For visualization clarity, Figure 2 skips such a case.
- When the pool is balanced, the price is adjusted to reflect price increase due to the increase of the demand. This is illustrated with $a = 100$, and $b = 100$ in Figure 2.

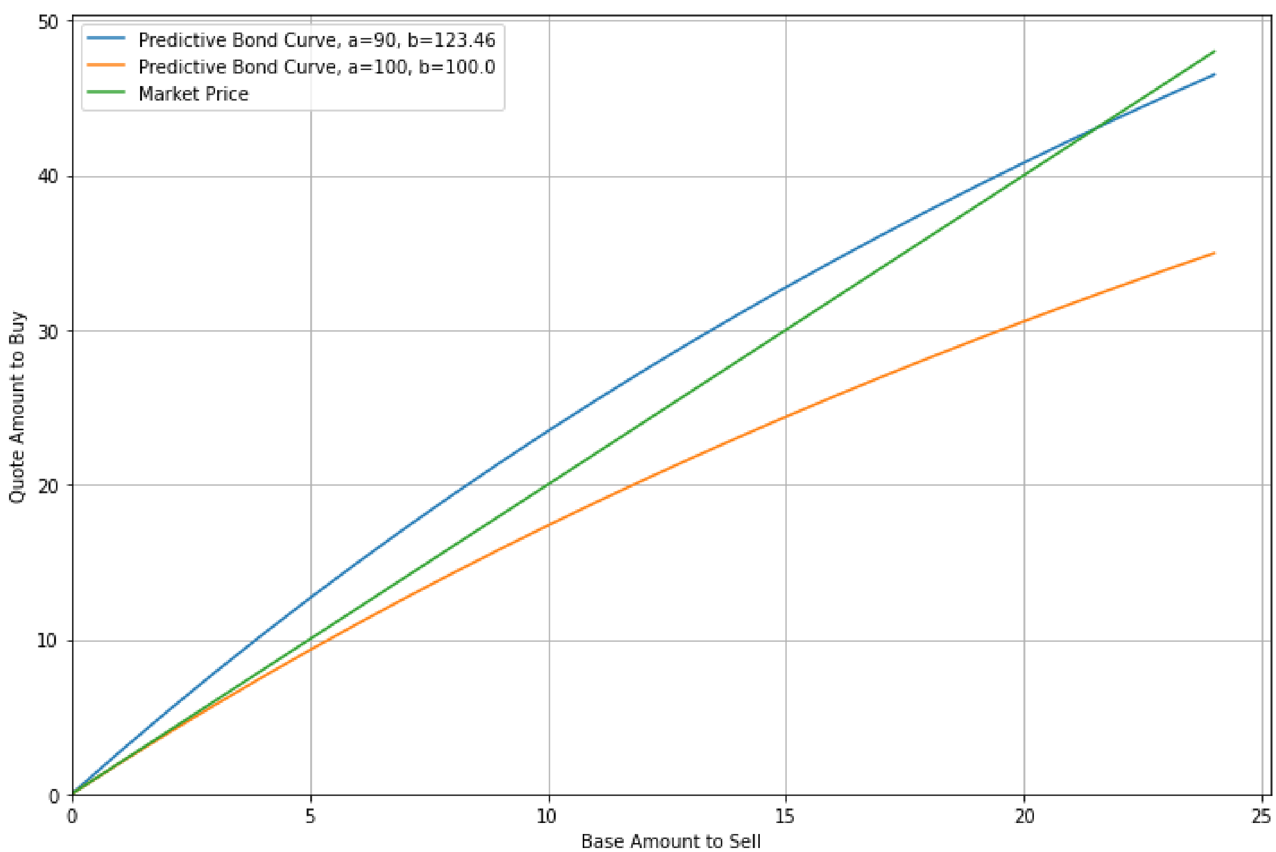

Figure 2. Bond curve showing the relation between the sell base amount and buy quote amount ($s = 1$, $P = 2$, $A = 100$, and $B = 100$).

Next, the bond curve price is visualized below. Quote amount per base token is the amount of quote token per base token to sell. When the pool is imbalanced, the price is adjusted around the market price accordingly. Figure 3 shows that:

- When the base reserve is smaller than the target reserve, the bond curve price is higher than the market price.
- When the pool is balanced, the quote amount decreases with the increase of base tokens to sell (i.e., quote tokens become more expensive).

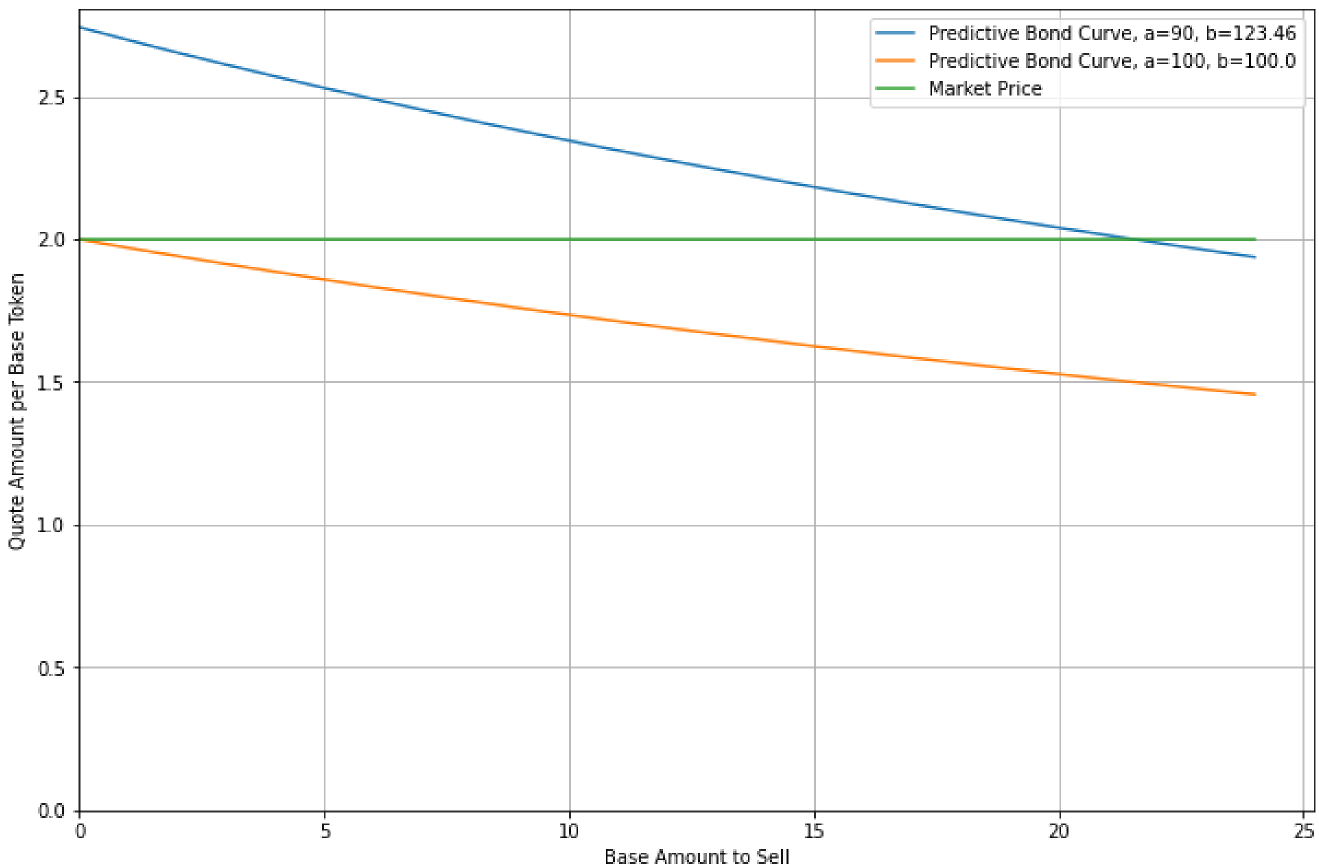

Figure 3. Price bond curve showing the relation between the sell base amount and quote amount per base token ($s = 1$, $P = 2$, $A = 100$, and $B = 100$).

4.2 Minimum Reserve Limitation

According to the above g(m), if $s < 1$, it is possible $g(m) > b$, which means there are less quote tokens in the pool than the trader buys. If the trader can never buy all the quote tokens, the quote tokens left after a swap can be any value close to 0. To prevent the situation of insufficient balance or being too unbalanced, we need to set a limitation on the minimum amount of reserve left in the pool.

Let the T be the constant that specifies the minimum reserve constraint, $0 < T < 1$.

if $b - g(m) > T \cdot B$:

swap out amount = $b - T \cdot B$

else:

swap out amount =$g(m)$

Assuming the oracle price is unchanged all the time, with $A = 100$, $B = 100$, the reserve amount change can be visualized as follows:

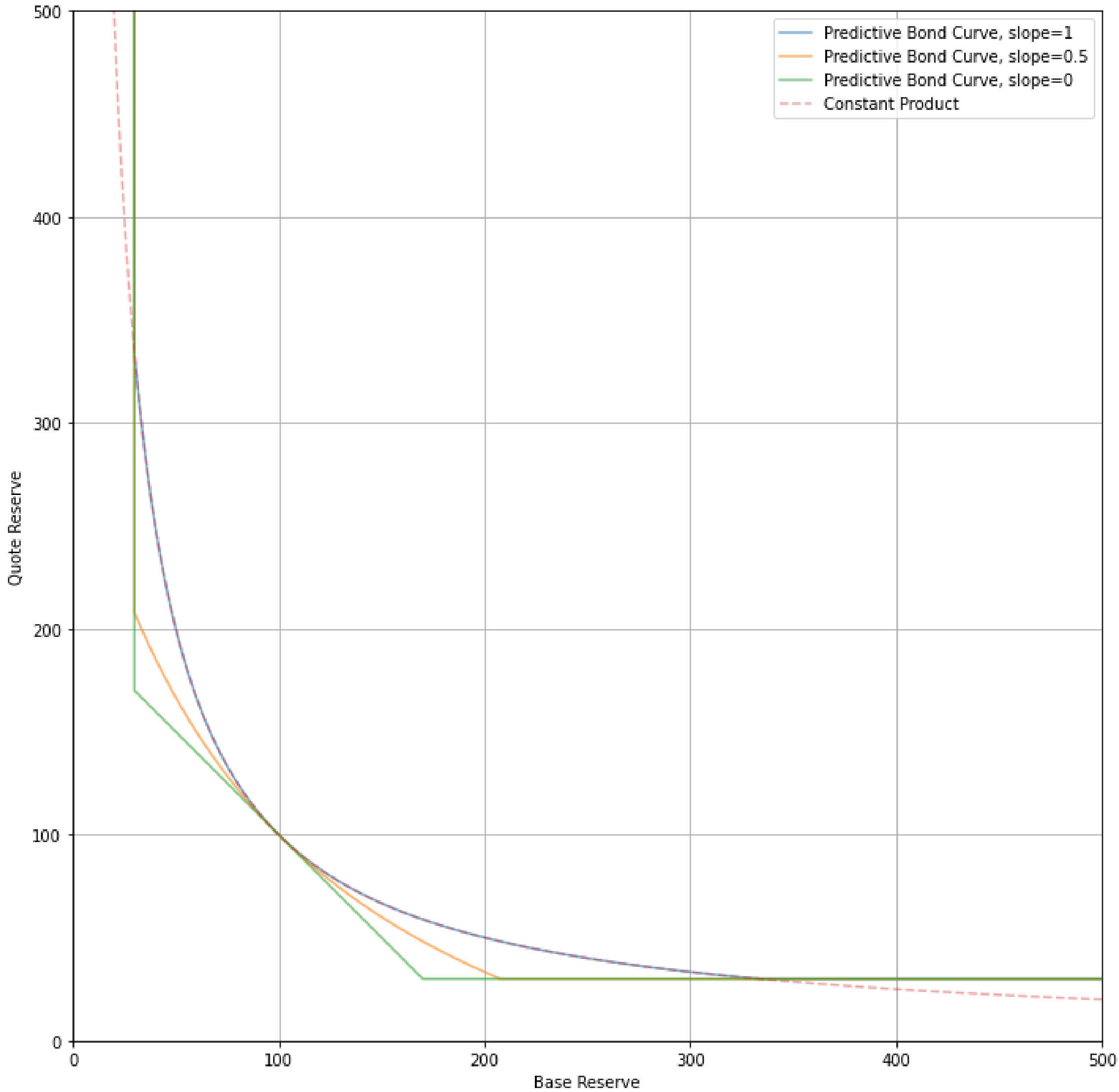

Figure 4. Bond curve comparison between different slopes and constant product. The dashed line shows the constant product curve. The constant product curve overlaps with Deltafi V2 curve when $s = 1$, unless the reserve is less than the limit.

Stable Swap. Figure 4 shows that slope adjustments support both stable swap and volatile assets.

5 Deposit and Withdrawal

5.1 The Problem

In the deposit/withdrawal mechanism in the traditional constant product formula, the ratio between the base and quote token deposited to the pool is the same as the pool's base/quote ratio. The base/quote ratio in the tokens to be withdrawn is also aligned with the pool's current base/quote ratio. Unless the liquidity provider is lucky enough to make the withdrawal under the same pool's base/quote ratio as when they made the deposit, they will suffer from impermeant loss from the changing ratio.

The source of impermanent loss is the change in the ratio between deposit and withdrawal. The proposed solution below is to resolve this problem.

5.2 Deposit

Suppose that the target reserves are A_0 and B_0. $\frac{A_0}{B_0}$is the target base/quote token ratio we want to maintain. The actual reserves are $A_1$and $B_1$and base oracle price = p_a quote oracle price = p_b. *Normalized reserves* A and B are defined as below:

$$A_{norm} = \frac{a \cdot p_a + b \cdot p_b}{A_0 \cdot p_a + B_0 \cdot p_b} \cdot A_0$$

$$B_{norm} = \frac{a \cdot p_a + b \cdot p_b}{A_0 \cdot p_a + B_0 \cdot p_b} \cdot B_0$$

From the above equations, let $\Delta a = a - A$; $\Delta b = a - B$. A_{norm}, B_{norm}, Δa , and Δb have the following properties:

1. $\Delta a \cdot p_a + \Delta b \cdot p_b = 0$
2. $\frac{A_{norm}}{B_{norm}} = \frac{A_0}{B_0}$
3. Whenever a deposit comes in, the new target base amount will be $A_0' = A_{norm} + d_a$, the new target quote amount will be $B_0' = B + d_b$, and Δb will be unchanged.
4. When a user makes a deposit, the user shares are calculated on both the base and quote sides, and we record the total share supplies of both base and quote:
    a. $share_a = \frac{d_a}{A_{norm}} \cdot Supply_a$; $Supply_a = Supply_a + share_a$
    b. $share_b = \frac{d_b}{B_{norm}} \cdot Supply_b$; $Supply_b = Supply_b + share_b$

5.3 Withdraw

Find A_1 and $B_1$with the following property.

$$\Delta a = a - A_1$$

$$\Delta b \ = \ b - B_1$$

$$\frac{A_0}{B_0} = \frac{A_1}{B_1}$$

$$\Delta a \ \geq 0$$

$$\Delta b \ \geq 0$$

$$\Delta a = 0 \ or \ \Delta b = 0$$

Since the formula is symmetric, we only consider one direction.

If $a > b \cdot \frac{A_0}{B_0}$:

$$A_1 = b \cdot \frac{A_0}{B_0}$$

$$B_1 = b$$

$$\Delta a = a - b \cdot \frac{A_0}{B_0}$$

$$\Delta b \ = \ 0$$

$$w_a = A_1 \cdot \frac{share_a}{Supply_a} + \Delta a \cdot tvlRatio$$

$$w_b = b \cdot \frac{share_a}{Supply_b}$$

$$A_0' = A_1 - w_a$$

$$B_0' = B_1 - w_a$$

The idea is to withdraw from A_1 and B_1 with the pool ratio until it uses up all the reserve on one end, then we only distribute the token on the other end based on the user TVL ratio.

TVL ratio refers to the ratio of user's deposit TVL to the total pool TVL at the time when the user withdraws. The formula for TVL ratio is:

$$TVL\ Ratio = \frac{share_a \cdot p_a + share_b \cdot p_b}{Supply_a \cdot p_a + Supply_b \cdot p_b}$$

5.4 Discussions

A groundbreaking feature of our share calculation for deposits is that we calculate separate shares for each side of the pool instead of a percentage of the whole pool. This guarantees that if we deposit and withdraw at the balanced point, which is the case most of the time, the withdraw base/quote token ratio is identical to the deposit ratio. The impermanent loss, by its definition, comes from a change in ratio between deposit and withdrawal. With this logic, we eliminate the ratio change so that we can eliminate the impermanent loss.

Theorem 3. If an LP makes a deposit with d_a amount of base token and d_b amount of quote token, and the shares received are $Share_a and\ Share_b$. At withdrawal time, the token amounts received are w_a and w_b for base and quote tokens, respectively. In a balanced pool state, $\frac{d_a}{d_b} = \frac{w_a}{w_b}$.

Proof. [See addendum]

From theorems 1, 2, and 3, it can be derived that the deposit and withdraw logic proposed here promotes fair share allocation.

6 Risk Management

This section discusses additional optimizations for higher pricing efficiency and mitigation solutions for oracle risks.

6.1 Front-Running

The solution is subject to the risk of being front-run — akin to a professional market maker that broadcasts its strategy to the market before executing its trades. Fortunately, DeltaFi's bond curve eliminates impermanent loss in pools despite the existence of arbitrage bots. Still, the following precautions are in place to further improve profitability.

- Front-running primarily occurs in extreme market conditions. Slot-by-slot oracle publishing limits such opportunities with our price correction techniques.
- Corrected oracle prices used in DeltaFi consider market volatility to mitigate front-running risks further.

6.2 Inventory Risk

Inventory risk comes with being overweight on one pair as the market moves against that pair. The bond curve design balances the trade pool automatically and avoids such an issue.

6.3 Oracle Manipulation

Oracles are a key attack vector for DeFi exploits through price manipulation. Such attacks are possible because DeFi protocols rely on one market price data source, which can be potentially manipulated. This is not the case for Pyth Network by aggregating price data from numerous data sources. Furthermore, the confidence interval gives additional signals if selected data sources are manipulated.

6.4 Token Availability

Furthermore, Pyth Network oracles are limited to larger-cap cryptocurrencies. To mitigate this issue, long-tail assets are supported with Serum prices as an oracle. Oracle manipulation can be an issue in such a case, which is true in any crypto exchange for the low liquidity. Fortunately, manipulation incentives are not high for low liquidity as well.

6.5 Oracle Failures

Due to Pyth Network or Solana system failures, Oracles may become stale or invalid. Failure detections have been implemented to prevent AMM from working when oracle prices are stagnant and have price anomalies such as large price deviations and too big confidence intervals.

6.6 Customized Pools

Professional market makers can leverage their in-house oracles and algorithms for parameter adjustments and more efficient pricing.

7 Evaluation Methodology

In our back-testing, we simulate traders' behavior of trading in ETH-USDC pool using real Binance and Pyth Network market price data, and compare different pools set up using the metrics. In this section, the evaluation methodology and data sets are detailed.

7.1 Retail & Professional Traders

The trading activities are modeled on arbitrager and retail trader bots.

- Arbitrage bots can trade in any amount to make a profit for themselves. An arbitrager searches for an excellent sell amount on either side and trades, assuming complete information about all arbitrage opportunities.
- Retail traders can trade a maximum of 10 ETH or 15000 USDC to simulate the normal retail trader behaviors. Retail traders will have a random trade direction and amount.
- Varying the percentage of arbitrageur and retail trader bots validates the predictive bond curve robustness.

7.2 Evaluation Cycles

We execute trader bots' logic in cycles. Each cycle has a point price that comes from real price history data, and several trader bots will execute trade logic based on the price point. There are configuration parameters of the cycle:

- Percentage of arbitrageur bots,
- Percentage of retail trader bots, and
- Trade direction probability.

7.3 History Price Dataset

The algorithms are back-tested with real Binance tick by tick price data and Pyth Network slot by slot price data.

- Binance price data: ETH/USDC data is tick by tick price from Feb 28, 2021 to Oct 27, 2021. This time period covers the flash crash in May 2021, normal fluctuation patterns, and price recovery from bear market. Binance price data is sampled every 10 seconds on average. Trades generated from arbitrager and retail trader bots are executed in random order at each price point.

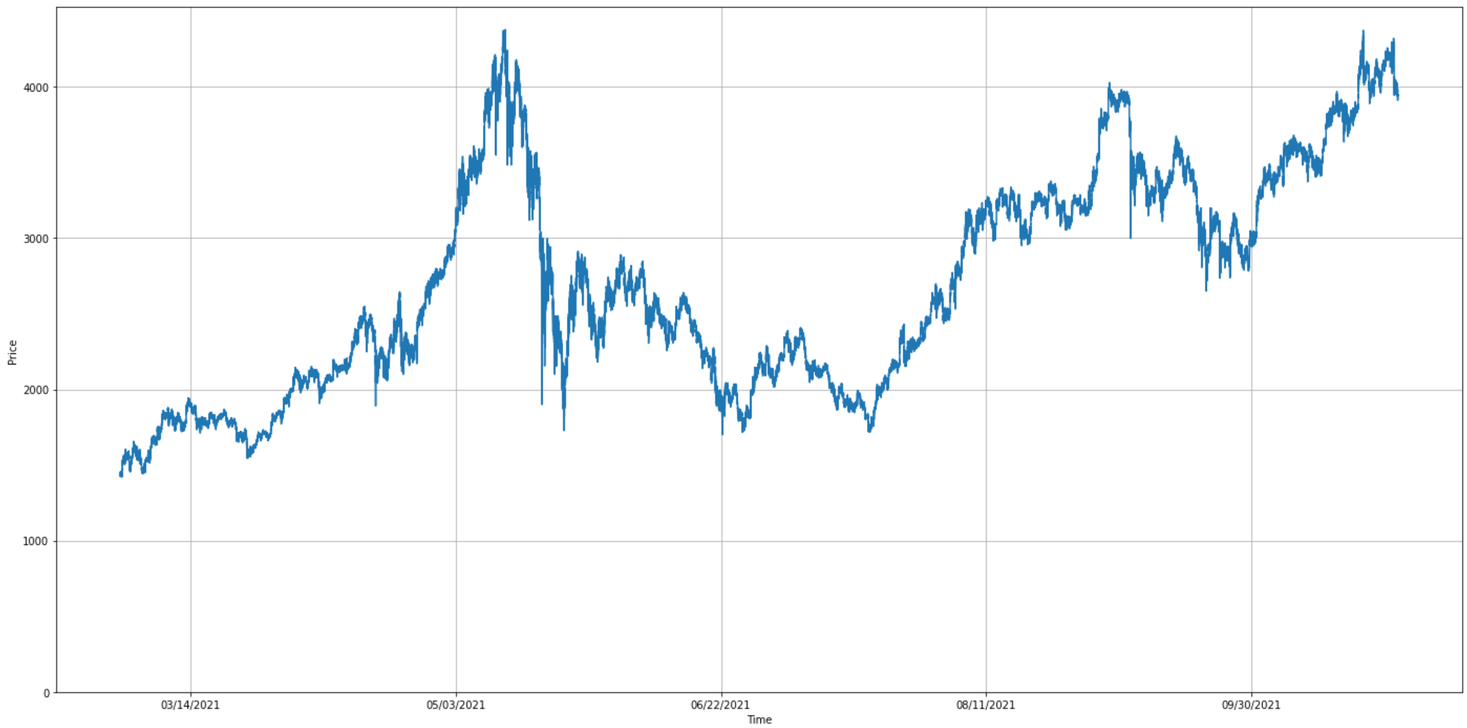

Figure 5. Binance tick by tick price data for ETH/USDC token pair from Feb 28, 2021 to Oct 27, 2021.

- Pyth Network data: ETH data is slot-by-slot price from Feb 3, 2022 to Mar 30, 2022. It includes all data fields such as current slot price, exponential moving average, time-weighted average price, last slot price, confidence interval, etc. For example, probability distributions for confidence intervals are constructed for back-testing. An exponential moving average is used for volatility back-testing.

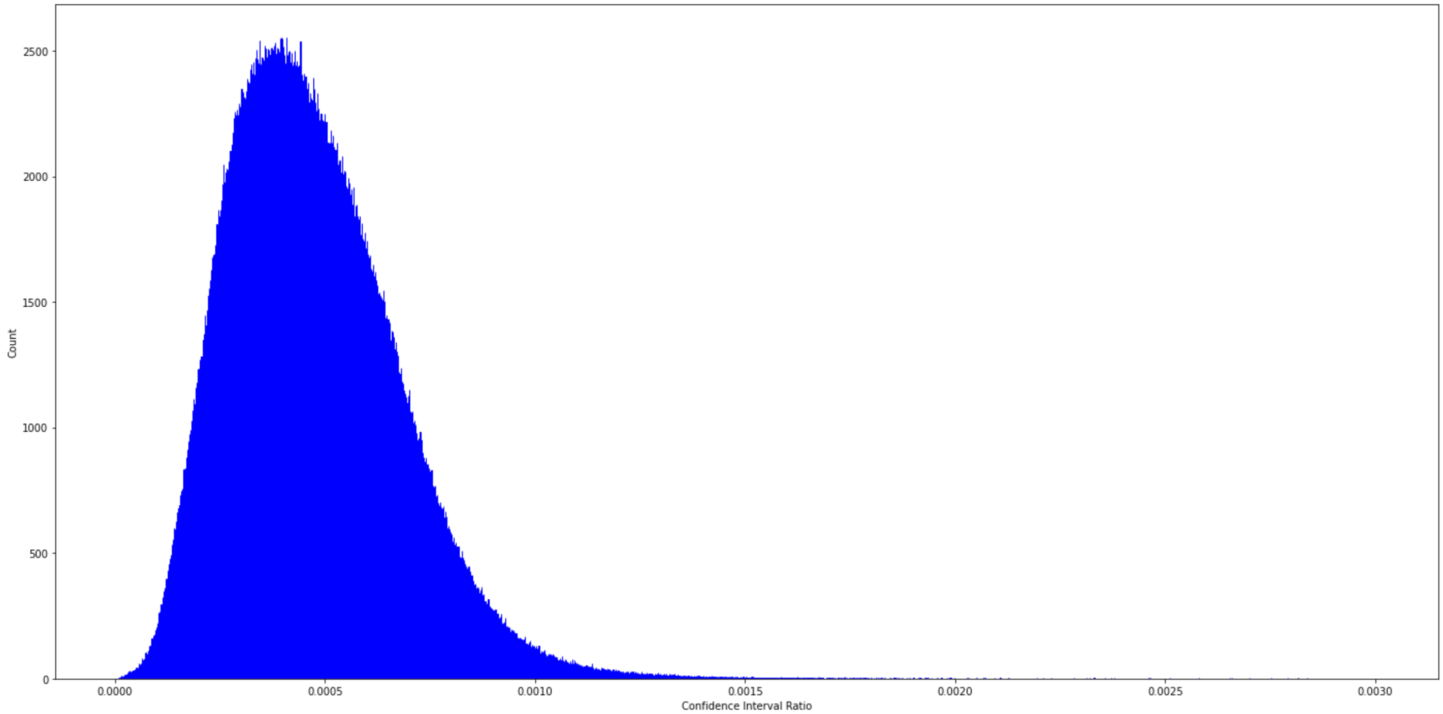

Figure 6. Confidence interval ratio distribution. It is adopted for confidence interval sampling in the back-testing. The confidence interval ratio is defined as the ratio between confidence interval and oracle price.

8 Evaluation Metrics

8.1 Price Impact

Price impact refers to the number of actual price deviations from the implied price. The average price impact level indicates the slippage level in general.

8.2 TVL Index

The TVL (total value locked) index refers to the ratio of the current TVL with the current market price to the initial pool reserve with the current market price. This metric is for monitoring the value change in our pool. Because TVL is related to the current market price of each token:

$$TVL_{current} = a \cdot P_a + b \cdot P_b,$$

where a is the current base reserve, b is the current quote reserve, P_a is the current base token price, and P_b is the current quote token price.

It is unfair to compare TVL values at the time of deposit and withdrawal since the price of base and quote tokens change over time. Therefore, TVL changes are compared based on the price point at each time point. Recall that A_0 is the target base reserve amount and B_0 is the target quote reserve amount. Impermanent loss can thus be calculated with:

$$TVL\ Index = \frac{a \cdot P_a + b \cdot P_b}{A_{init} \cdot P_a + B_{init} \cdot P_b}.$$

In back-testing, the TVL index is evaluated at each time point to see the impact of impermanent loss.

8.3 Capital Efficiency

Capital efficiency refers to how fast our pool can grow in TVL and let liquidity providers get profit. The TVL index at a single time point implies the TVL profit. Therefore, the TVL index is reused for capital efficiency evaluation.

8.4 TVL Gain Rate

TVL gain measures the gain or loss of deposit-withdraw behaviors in our pool. Suppose d_a base token and d_b quote token is deposited at the time t_0. At the time t_1, the same liquidity provider withdraws all the deposits, with w_a base token and w_b quote token. The TVL gain for this particular deposit and withdrawal is defined as:

$$TVL\ Gain = \frac{w_a \cdot P_a + w_b \cdot P_b}{d_a \cdot P_a + d_b \cdot P_b} - 1.$$

TVL gain shows how individual liquidity providers profit from the pool, while the TVL index measures the capital efficiency of the whole pool.

The predictive bond curve guarantees pool value increases in the long run. However, it does not assure that each individual can make a profit if the deposit and withdrawal mechanism is unfair. To this end, the TVL gain rate is defined as TVL gain divided by the deposit time duration:

$$TVL\ Gain\ Rate = \frac{\frac{w_a \cdot P_a + w_b \cdot P_b}{d_a \cdot P_a + d_b \cdot P_b} - 1}{t_1 - t_0}.$$

The distribution of TVL gain rate with random liquidity provision bots tells how consistent an LP can profit.

9 Evaluation Results

In this section, comprehensive evaluation results are provided. More evaluation results are provided in the Addendum. The predictive bond curve is robust to prevent impermanent loss with similar results under various arbitrage bot percentages, liquidity depth, and trade direction probabilities. Therefore, this paper shows back-testing results for the default settings for illustration purposes.

9.1 Default Settings

Unless otherwise mentioned, the default settings are:

Metrics	**Value**
Arbitrage Bot %	20
Retail Trader %	80

Initial Base Reserve (USDC)	1427610
Initial Quote Reserve (ETH)	1000
Slope	1
Probability of Selling Base	50%
Probability of Sell Quote	50%
Fee	0.0%
Oracle Correction	disabled

Table 2. Default Evaluation Settings

9.2 Price Impact

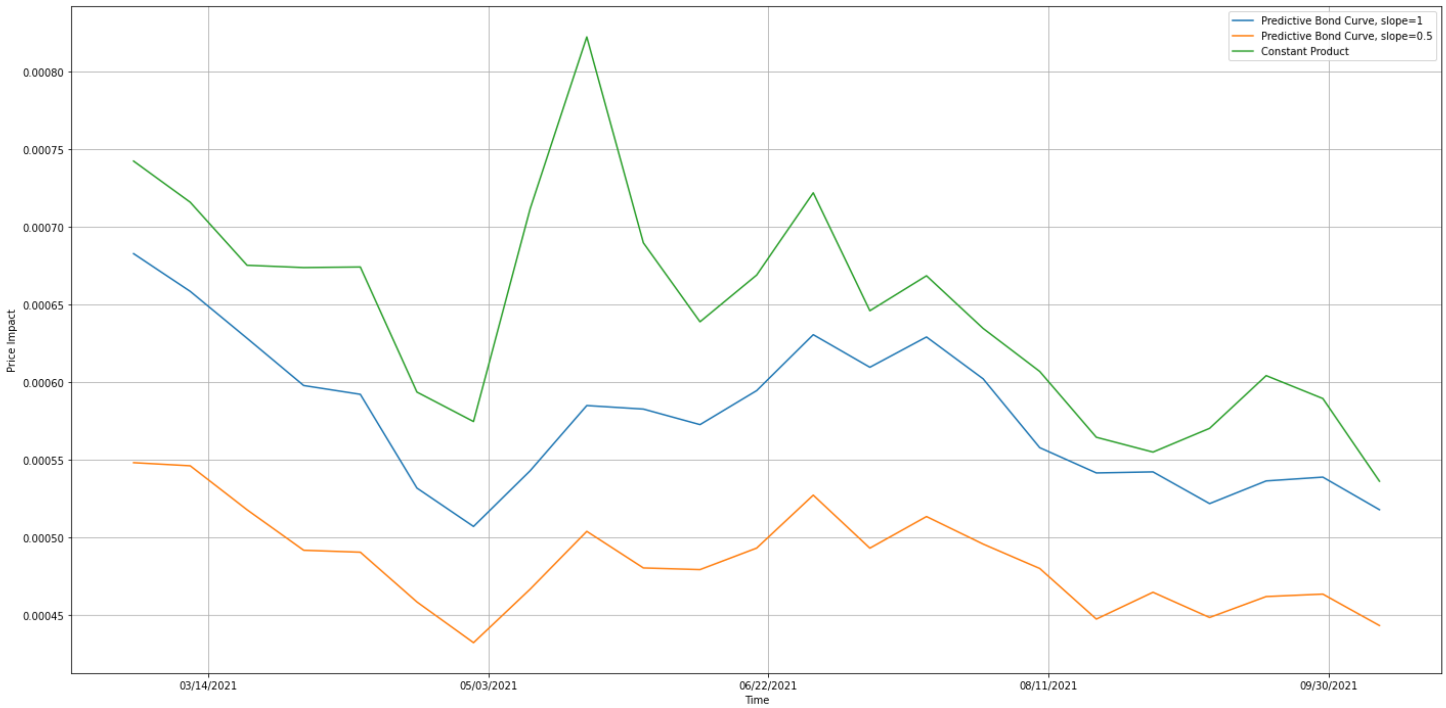

Figure 7. The predictive bond curve offers flexible price impact and improves capital efficiency without sacrificing impermanent loss by varying slopes. On the other hand, the constant product curve suffers from the highest price impact

9.3 TVL Index

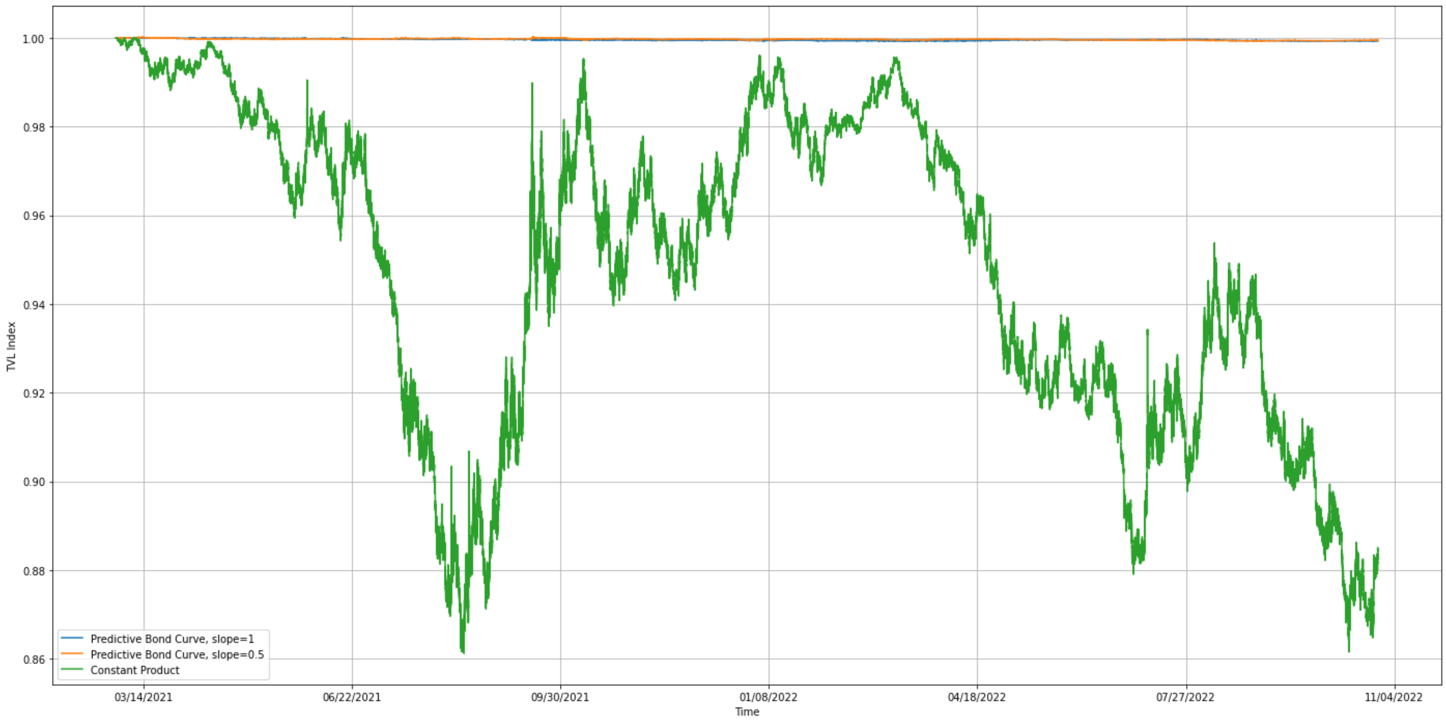

Figure 8. The predictive bond curve has a TVL of around 1 to eliminate permanent loss due to pool rebalancing incentives.

9.4 Low Liquidity

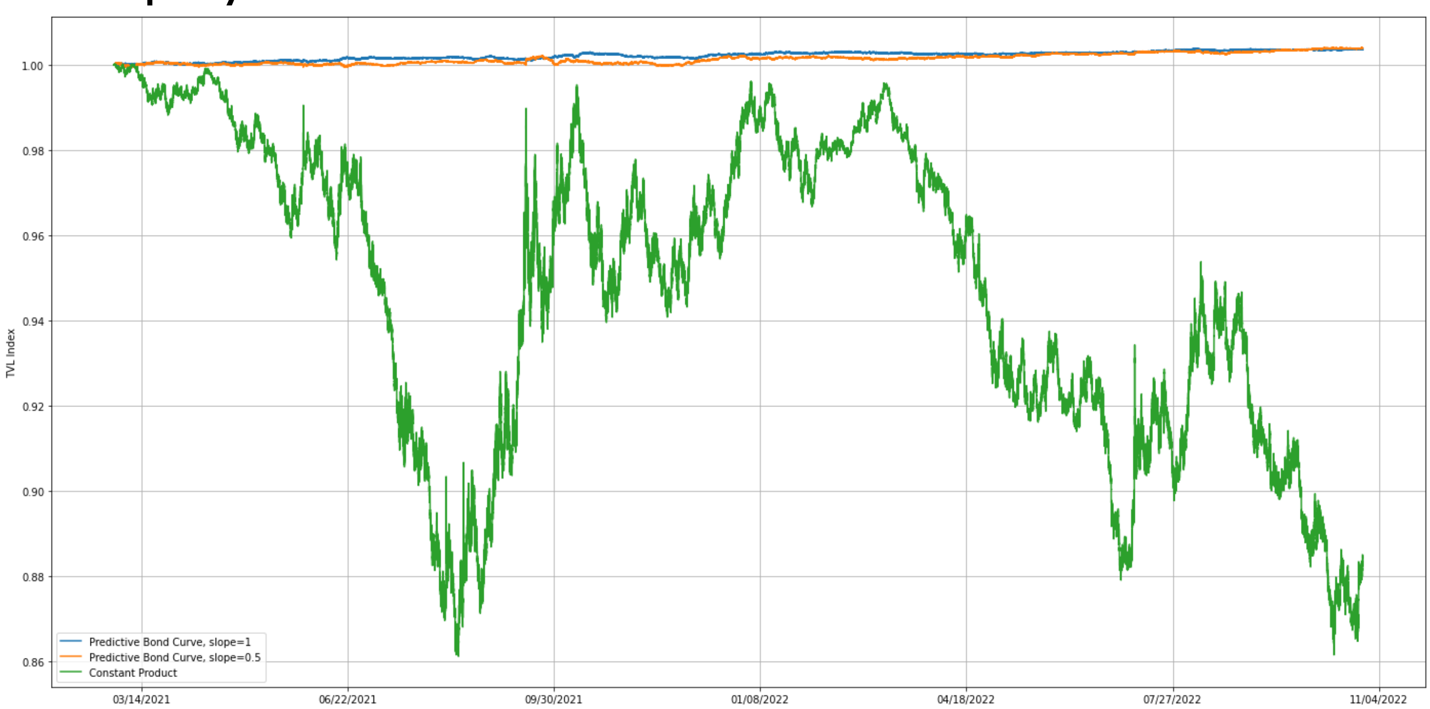

Figure 9. With 10% of the default liquidity, the predictive bond curve still mitigates the impermanent loss.

9.5 Irrational Trades

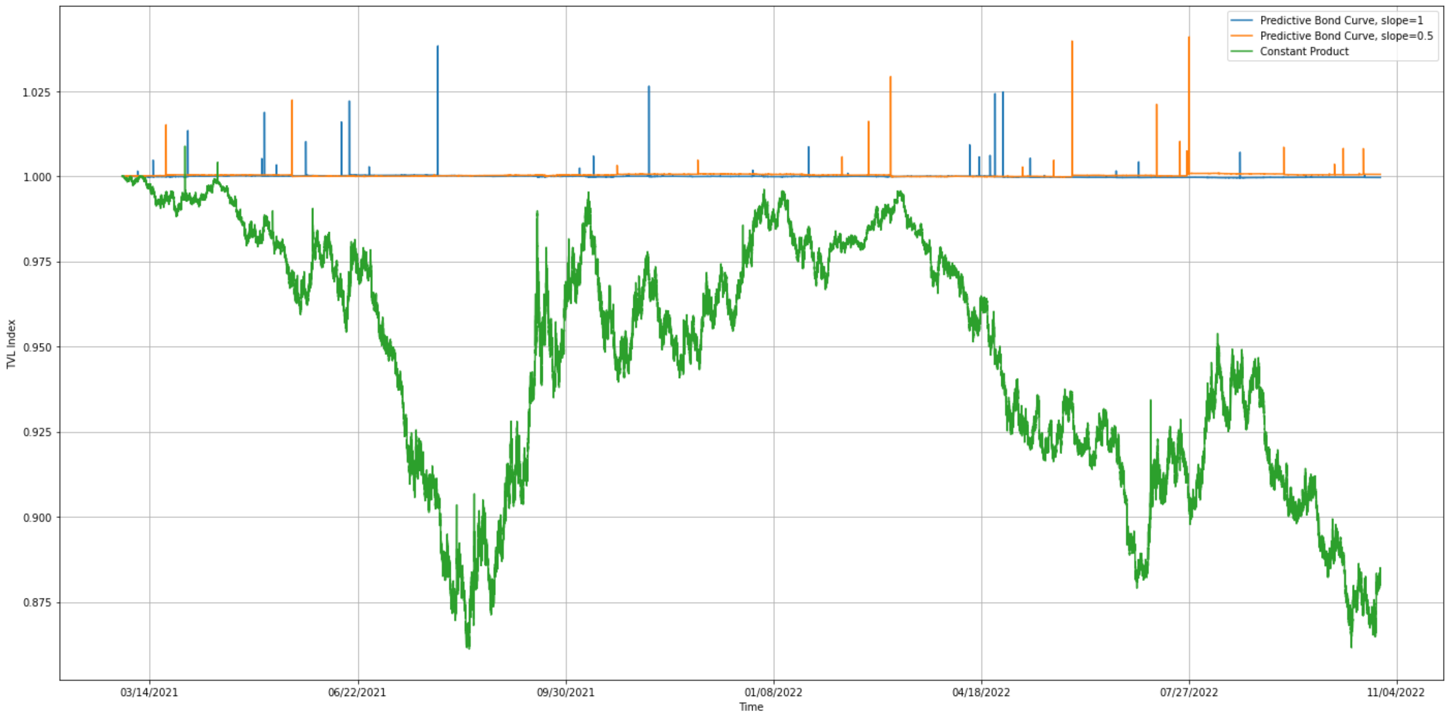

Figure 10. Irrational trade evaluation with random large trades. The large trade is up to 20% of the pool reserve. The rebalancing mechanism performs well under such extreme situations.

9.6 Deposit and Withdraw

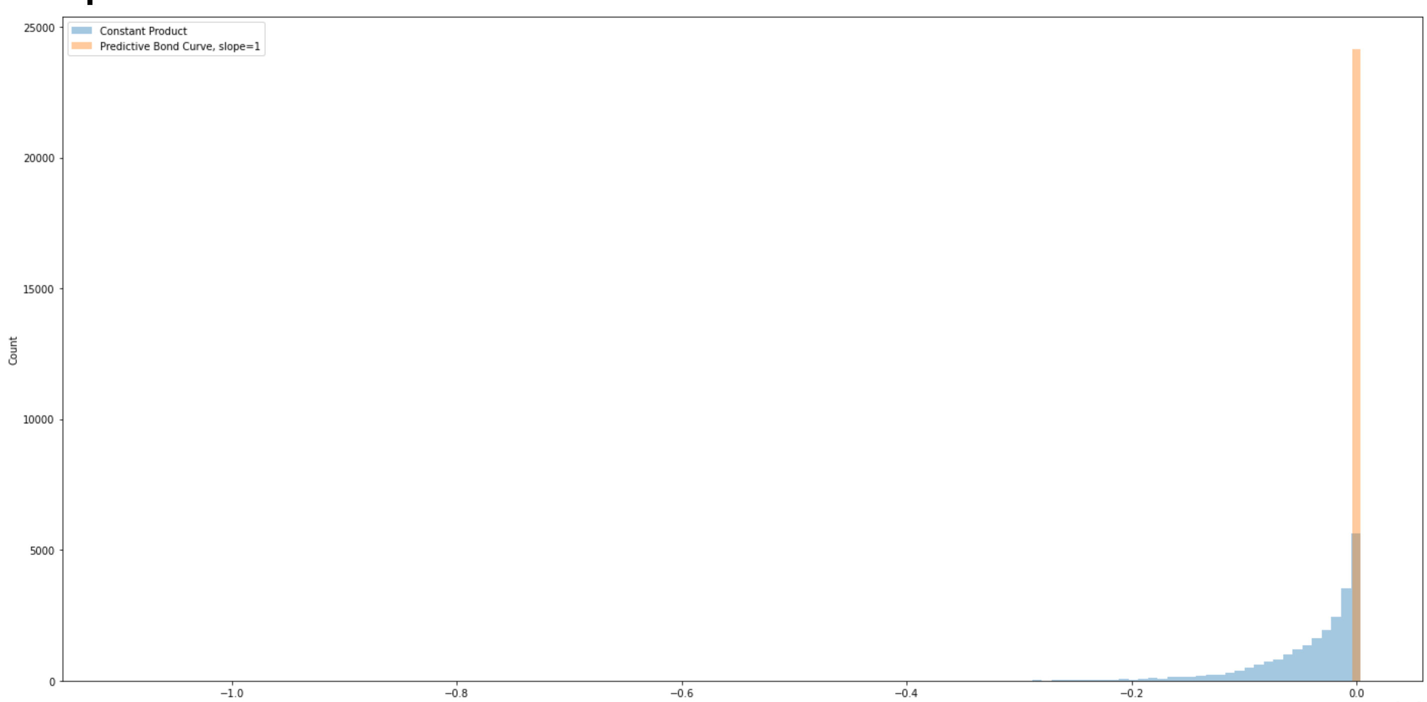

Figure 11. Randomly deposit and then withdraw after at least 10000 cycles (approximately 1 day). It shows individual deposits in the predictive bond curve pool retain their value, while deposits in the constant product curve pool are exposed to impermanent loss.

9.7 Trading Volume

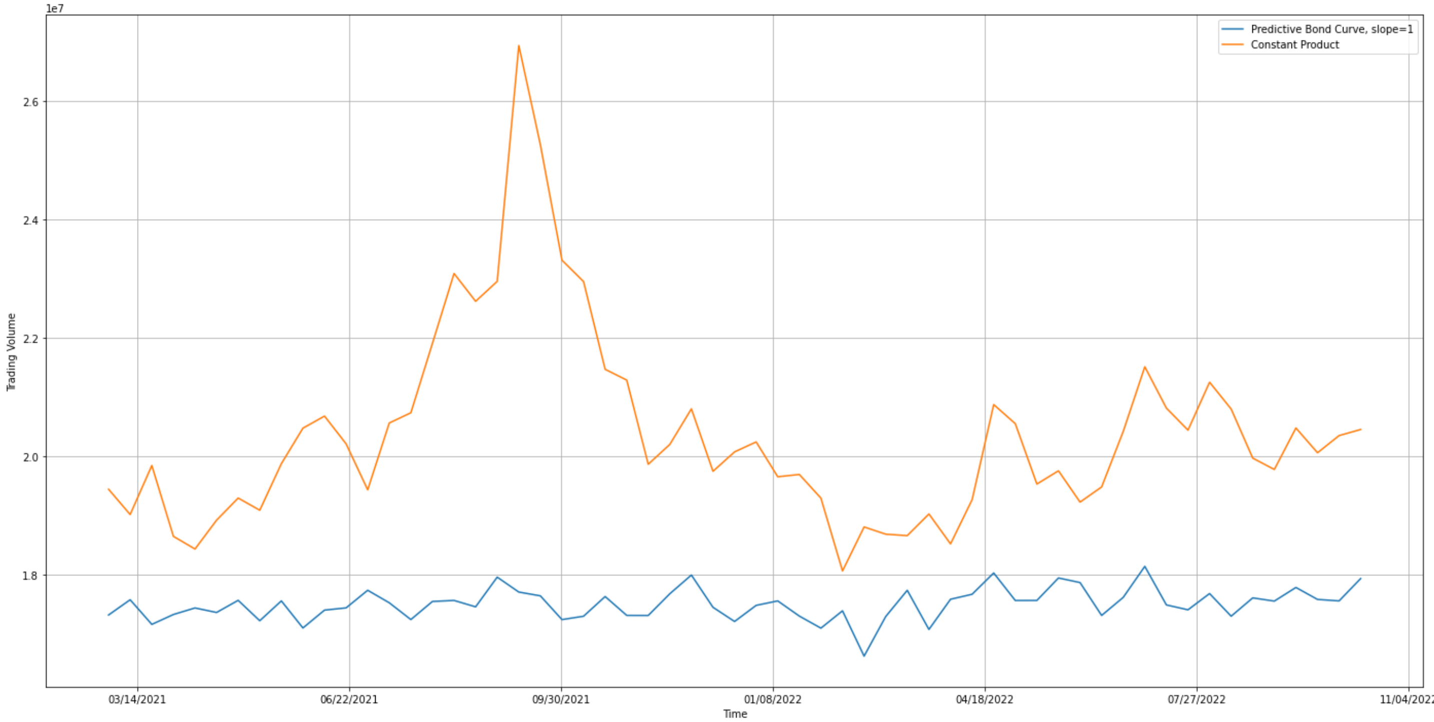

Figure 12. Trading volume is in the same order, though slightly lower for predictive bond curve. This is for the elimination of toxic flow.

9.8 Worst Scenario Analysis

The worst scenario is when the trading happens in one direction. In reality, this may happen during sudden price increases or drops. For such an analysis, the probability of selling ETH is set to 90%, while the other direction's probability is 10%.

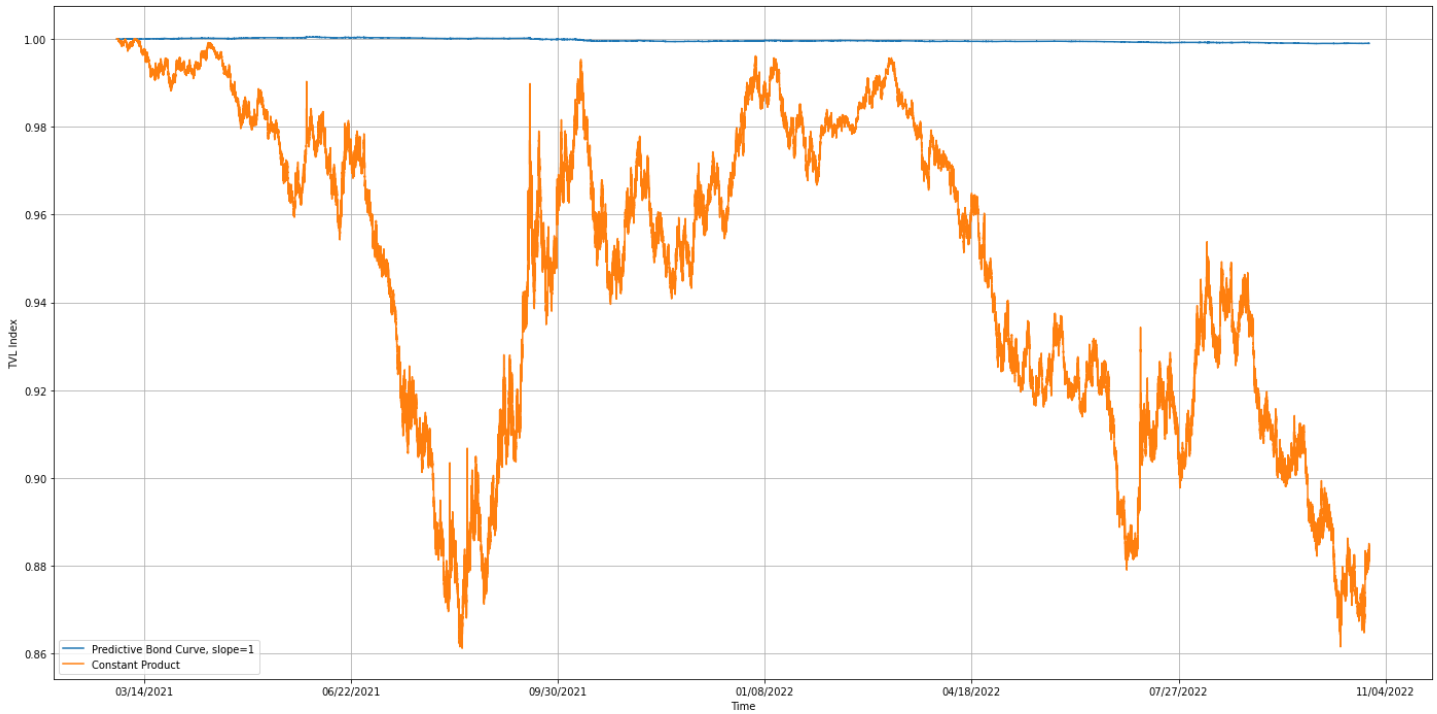

Figure 13. The predictive bond curve is robust to biased trading directions. It eliminates impermanent loss.

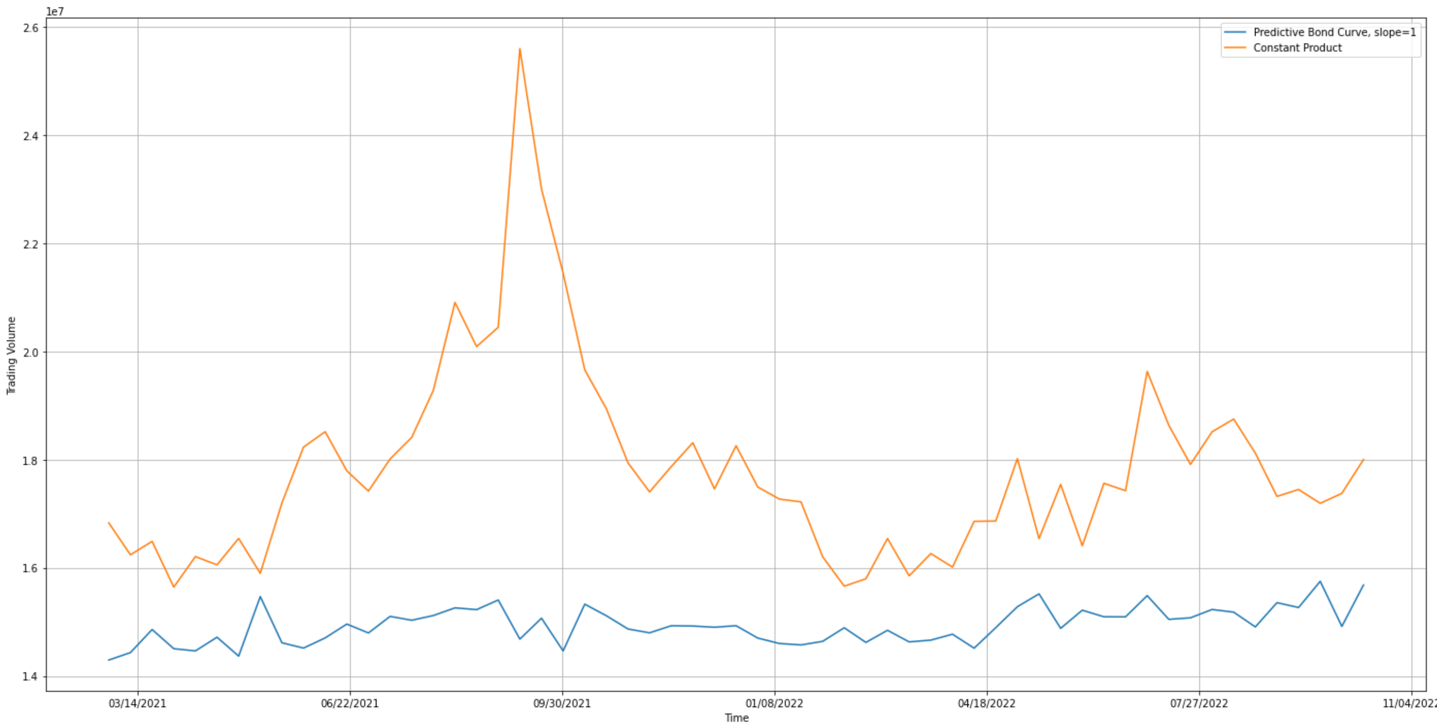

Figure 14. Trading volume is lower with biased trading directions for unfavorable prices. The volume is calculated periodically (i.e., every 40000 evaluation cycles).

9.9 Transaction Fee

Evaluation with 0.1% fee is to demonstrate the impact of the fee on TVL growth and trading volume. In addition, the Oracle correction option is added for comparison.

Figure 15. Both bond curves show TVL increases due to the fee. Constant product is not stable with loss in TVL. During price fluctuations, the constant product curve suffers a lot of liquidity loss.

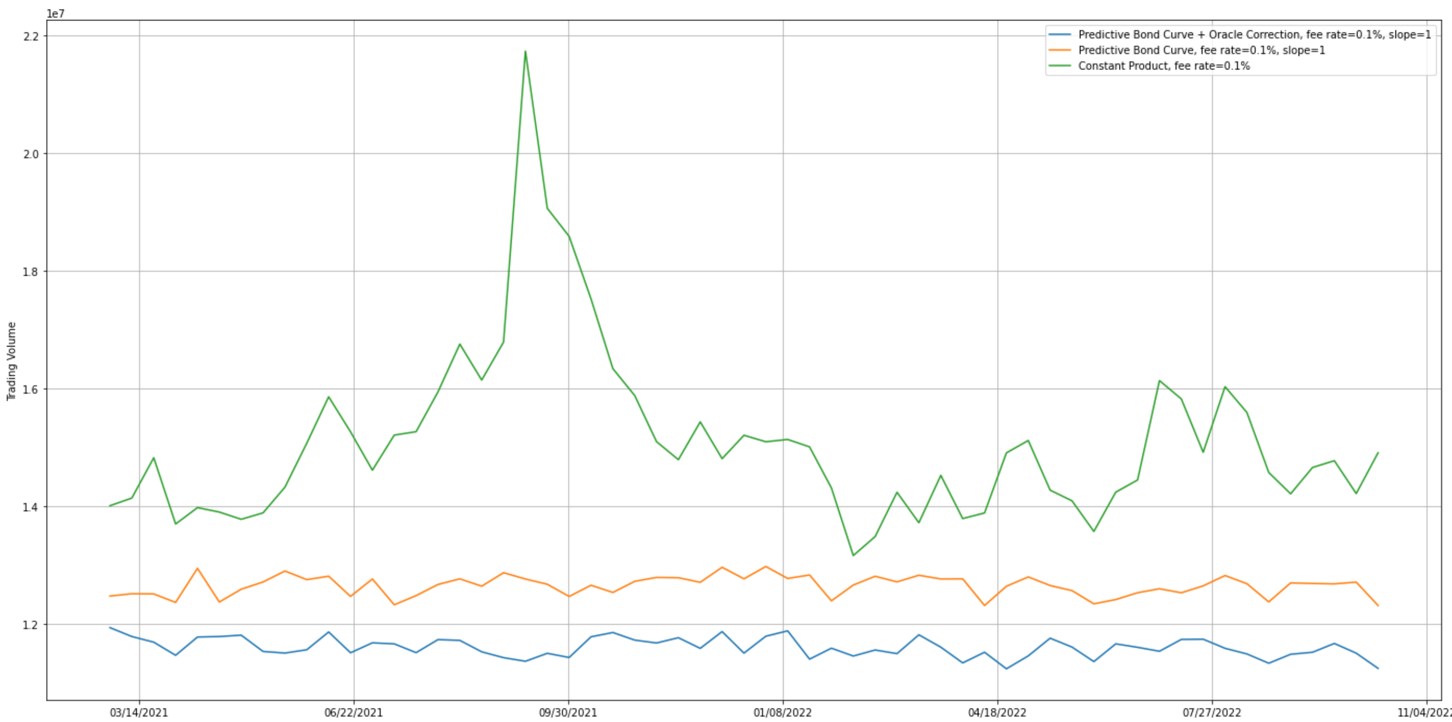

Figure 16. Trading volume is reduced with fees for blocking some arbitrage trades.

10 Conclusion

DeltaFi proposes predictive bond curve to eliminate impermanent loss in automated market maker. It innovates the AMM design by solving the issues faced by the constant product formula. The predictive bond curve is customizable to improve capital efficiency and slippage. It can apply to both volatile assets and stable swaps. Comprehensive back-testing results validate the solution with real Binance and Pyth Network price data.

11 Addendum

11.1 Differential Equation Solution for g(m)

$$g(m) = \frac{P \cdot A}{B} \cdot \int_0^m \frac{s \cdot \big(b - g(x)\big) + (1-s) \cdot B}{s \cdot (a+x) + (1-s) \cdot A}\, dx$$

$$g'(m) = \frac{P \cdot A}{B} \cdot \frac{s \cdot \big(b - g(m)\big) + (1-s) \cdot B}{s \cdot (a+m) + (1-s) \cdot A}$$

$$(a+m) \cdot g'(m) + \frac{1-s}{s} \cdot A \cdot g'(m) + \frac{P \cdot A}{B} \cdot g(m) = \frac{P \cdot A}{B} \cdot \left(b + \frac{1-s}{s} \cdot B\right)$$

$let\ t\ =\ a+m,\ g_0(t) = g(t-a) = g(m),\ then\ g_0'(t) = g'(m)$

$$\left(t + \frac{(1-s) \cdot A}{s}\right) \cdot g_0'(t) + \frac{P \cdot A}{B} \cdot g_0(t) = \frac{P \cdot A}{B} \cdot \left(b + \frac{1-s}{s} \cdot B\right)$$

$let\ r = t + \frac{(1-s) \cdot A}{s},\ g_1(r) = g_0\left(r - \frac{(1-s) \cdot A}{s}\right) = g_0(t),\ g_1'(r) = g_0'(t)$

$$r \cdot g_1'(r) + \frac{P \cdot A}{B} \cdot g_1(r) = \frac{P \cdot A}{B} \cdot \left(b + \frac{1-s}{s} \cdot B\right)$$

$$r^{\frac{P \cdot A}{B}} \cdot g_1'(r) + \frac{P \cdot A}{B} \cdot r^{\frac{P \cdot A}{B} - 1} \cdot g_1(r) = \frac{P \cdot A}{B} \cdot \left(b + \frac{1-s}{s} \cdot B\right) \cdot r^{\frac{P \cdot A}{B} - 1}$$

$$\frac{d}{dt}\left(r^{\frac{P \cdot A}{B}} \cdot g_1(r)\right) = \frac{P \cdot A}{B} \cdot \left(b + \frac{1-s}{s} \cdot B\right) \cdot r^{\frac{P \cdot A}{B} - 1}$$

$$r^{\frac{P \cdot A}{B}} \cdot g_1(r) = \left(b + \frac{1-s}{s} \cdot B\right) \cdot r^{\frac{P \cdot A}{B}} + c$$

$$g_1(r) = b + \frac{1-s}{s} \cdot B + \frac{c}{r^{\frac{P \cdot A}{B}}}$$

$$g_0(t) = b + \frac{1-s}{s} \cdot B + \frac{c}{\left(t + \frac{(1-s) \cdot A}{s}\right)^{\frac{P \cdot A}{B}}}$$

$$g(m) = b + \frac{1-s}{s} \cdot B + \frac{c}{\left(a + m + \frac{(1-s) \cdot A}{s}\right)^{\frac{P \cdot A}{B}}}$$

Since g(m) is a result of integration from 0 to m, we know that:

$$g(0) = 0$$

$$c = -\left(b + \frac{1-s}{s} \cdot B\right) \cdot \left(a + \frac{(1-s) \cdot A}{s}\right)^{\frac{P \cdot A}{B}}$$

$$g(m) = \left(b + \frac{1-s}{s} \cdot B\right) \cdot \left(1 - \left(\frac{s \cdot a + (1-s) \cdot A}{s \cdot (a+m) + (1-s) \cdot A}\right)^{\frac{P \cdot A}{B}}\right)$$

11.2 Finding Balanced Reserves

Given the current base and quote reserves a, b, and the last updated target reserve A_0, B_0, we can find the balanced point A, B.

Let the current reserve is a result of selling m_0 base tokens, according to the $g(m)$ formula:

$$B - b = \frac{B}{s} \cdot \left(1 - \left(\frac{A}{A + s \cdot m_0}\right)^{\frac{P \cdot A}{B}}\right)$$

$$A + m_0 = a$$

Then,

$$b = B - \frac{B}{s}\left(1 - \left(\frac{A}{A + s \cdot m_0}\right)^{\frac{P \cdot A}{B}}\right)$$

With $\frac{A}{B} = \frac{A_0}{B_0}$,

$$b = \frac{B_0}{A_0} \cdot \left(1 - \frac{1}{s}\right) \cdot (a - m_0) + \frac{B_0}{s \cdot A_0} \cdot (a - m_0) \cdot \left(\frac{a - m_0}{a - m_0 + s \cdot m_0}\right)^{\frac{P \cdot A}{B}}$$

Let $f(x) = \frac{B_0}{A_0} \cdot \left(1 - \frac{1}{s}\right) \cdot (a - x) + \frac{B_0}{s \cdot A_0} \cdot (a - x) \cdot \left(\frac{a-x}{a-x+s \cdot x}\right)^{\frac{P \cdot A}{B}} - b$

This is a monotonically decreasing function, with limited decimals of any token, we can get the result using a binary search that finds the solution of $f(x) = 0$, which is the desired m_0 we want.

11.3 Proof of Theorem 1

When selling m base token, we get quote token. We assume that the market price is at P, therefore the net gain is $f(m) = g(m) - P \cdot m$

Let m_0 be the sell amount that lets the pool back to a balanced ratio, we have:

$$\frac{s\cdot\left(b-g(m_0)\right)+(1-s)\cdot B}{s\cdot(a+m)+(1+s)\cdot A}=\frac{B}{A}$$

From the formula of $g(m)$, it is evident that it is a monotonically increasing function to m

For any m, $f(m_0)-f(m)=g(m_0)-p\cdot m_0-g(m)+p\cdot m=g(m_0)-g(m)+p(m-m_0)$

From the original definition of g(m):

$$g(m_0)-g(m)=\frac{P\cdot A}{B}\cdot\int_0^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}dx-\frac{P\cdot A}{B}\cdot\int_0^{m}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot B}dx$$

$$g(m_0)-g(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}dx$$

Then:

$$f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}dx-P\cdot(m_0-m)$$

$$f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}dx-P\cdot\int_m^{m_0}1dx$$

$$f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}-\frac{B}{A}dx$$

From the monotonic of $g(m)$ and the definition of m_0 and $g(m_0)$, we know that:

If $m>m_0$, for any$x\in(m_0,m)$, $\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}-\frac{B}{A}<0$, therefore

$$f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}-\frac{B}{A}dx>0$$

If $m<m_0$, for any $f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}-\frac{B}{A}dx>0$

Therefore,

$$f(m_0)-f(m)=\frac{P\cdot A}{B}\cdot\int_m^{m_0}\frac{s\cdot\left(b-g(x)\right)+(1-s)\cdot B}{s\cdot(a+x)+(1-s)\cdot A}-\frac{B}{A}dx>0$$

From the above, we have $f(m_0)-f(m)>0$, therefore, for any m, $f(m_0)-f(m)>0$, which means with sell amount m_0, the user's profit is larger than any other sell amount

11.4 Proof of Theorem 2

Let $m_1,m_2\ldots m_n$ be a sequence of n small selling amount of base token, $m_i$can be either positive or negative to represent actual buy or sell. Let $M=\sum_{i=0}^{n}m_i$

Treat a and b as variables in the g(m) curve we have, let

$$g_0(a,b,m) = \frac{P \cdot A}{B} \cdot \int_0^m \frac{s \cdot \big(b - g(x)\big) + (1-s) \cdot B}{s \cdot (a+x) + (1-s) \cdot A}\, dx$$

After each trade, the reserve changes. Let (a_0, b_0) be the initial reserve, and (a_i, b_i) is the reserve after i^{th}trade. The number of quote tokens we get from i^{th} trader is $g_0(a_i, b_i, m_i)$

For each pair of adjacent transactions with sell base amount $m_i\ and\ m_{i+1}$, we have:

$$a_i = a_{i-1} + m_{i-1}; b_i = b_{i-1} - r_{i-1}$$

From the definition:

$$\begin{aligned} g_0(a_{i-1}, b_{i-1}, m_i) + g_0(a_i, b_i, m_{i+1}) &= \frac{P \cdot A}{B} \\ &\cdot \left(\int_0^{m_i} \frac{s \cdot \big(b_{i-1} - g_0(a_{i-1}, b_{i-1}, x)\big) + (1-s) \cdot B}{s \cdot (a_{i-1} + x) + (1-s) \cdot A}\, dx \right. \\ &\left. + \int_0^{m_{i+1}} \frac{s \cdot \big(b_i - g_0(a_i, b_i, x)\big) + (1-s) \cdot B}{s \cdot (a_i + x) + (1-s) \cdot A}\, dx \right) \end{aligned}$$

Therefore,

$$\begin{aligned} g_0(a_{i-1}, b_{i-1}, m_i) + g_0(a_i, b_i, m_{i+1}) &= \frac{P \cdot A}{B} \\ &\cdot \left(\int_0^{m_i} \frac{s \cdot \big(b_{i-1} - g_0(a_{i-1}, b_{i-1}, x)\big) + (1-s) \cdot B}{s \cdot (a_{i-1} + x) + (1-s) \cdot A}\, dx \right. \\ &\left. + \int_{m_i}^{m_i + m_{i+1}} \frac{s \cdot \big(b_{i-1} - g_0(a_{i-1}, b_{i-1}, x)\big) + (1-s) \cdot B}{s \cdot (a_{i-1} + x) + (1-s) \cdot A}\, dx \right) \end{aligned}$$

From the property of integration,

$$g_0(a_{i-1}, b_{i-1}, m_i) + g_0(a_i, b_i, m_{i+1}) = g_0(a_{i-1}, b_{i-1}, m_i + m_{i+1})$$

Applying such property to the sequence of trade:

$$\sum_{i=1}^{n} g_0(a_{i-1}, b_{i-1}, m_i) = g_0\left(a_0, b_0, \sum_{i=1}^{n} m_i\right) = g_0(a_0, b_0, M)$$

Therefore, the total amount of quote tokens from the trade sequence equals the single trade.

11.5 Proof of Theorem 3

In order to prove Theorem 3, we need to prove the following lemma first

Lemma. At each moment, let A and B are the balanced base and quote reserve, $Supply_a and\ Supply_b$ are the total share supply of base and quote token, $Supply_a and\ Supply_b$

$\frac{Supply_a}{A} = \frac{Supply_b}{B}$ is always true.

Proof for **lemma:**

Initially, we set $Supply_a = A$; $Supply_b = B$, the condition is true

To prove the statement, we prove the following three sub-statements:

1. If at the current pool state, $\frac{Supply_a}{A} = \frac{Supply_b}{B}$is true, after a deposit, it is still true
2. If at the current pool state, $\frac{Supply_a}{A} = \frac{Supply_b}{B}$is true, after a withdrawal, it is still true
3. If at the current pool state, $\frac{Supply_a}{A} = \frac{Supply_b}{B}$is true, after a swap, it is still true

For statement 1:

Let d_a and d_b be the deposit amount of base and quote token, $share_a\ and\ share_b$ be the share issued to the user, $A_{norm}\ and\ B_{norm}$ are the normalized base and quote reserves, according to our method:

$$share_a = \frac{d_a}{A_{norm}} \cdot Supply_a$$

$$share_b = \frac{d_b}{B_{norm}} \cdot Supply_b$$

Then,

$$\frac{Supply_a + share_a}{A_{norm} + d_a} = \frac{Supply_b + share_b}{B_{norm} + d_b}$$

$A_{norm} + d_a\ and\ B_{norm} + d_b$ are the updated target reserves. Because target reserves and balanced reserves have the same ratio by definition

For statement 2:

Let $share_a\ and\ share_b$ be the shares to withdrawal, $w_a\ and\ w_b$ be the withdrawal amount of each side. Then, using our method, let $w_a'\ and\ w_b'$ be the actual decrease in balanced reserve, $A_0'\ and\ B_0'$ are the new target reserves calculated after this withdrawal.

From the withdrawal method, $A_0' = \frac{Supply_a - share_a}{Supply_a} \cdot A_1$, and $B_0' = \frac{Supply_b - share_b}{Supply_b} \cdot B_1$

Therefore, $\frac{A_0'}{Supply_a - share_a} = \frac{A_1}{Supply_a}$ and $\frac{B_0'}{Supply_b - share_b} = \frac{B_1}{Supply_b}$

By definition, $\frac{A_1}{B_1} = \frac{A}{B'}$,

Then, $\frac{Supply_a - share_a}{A_0'} = \frac{Supply_a}{A}$ and $\frac{Supply_b - share_b}{B_0'} = \frac{Supply_b}{B}$

Because target reserves and balanced reserves have the same ratio by definition, statement 2 is true.

***For statement 3*:**

Swap operation does not change Supply amounts and $\frac{A}{B}$, therefore, the statement is true

Since we only have three types of functions in our pool, when all the sub-statements are true, the lemma is true all the time

If the LP withdraws at a balanced state in our withdrawal method, $A_1 = A \; and \; B_1 = B$

Then $w_a = \frac{share_a}{Supply_a} \cdot A$ and $w_b = \frac{share_b}{Supply_b} \cdot B$

Then $\frac{w_a}{w_b} = \frac{\frac{share_a}{Supply_a} \cdot A}{\frac{share_b}{Supply_b} \cdot B} = \frac{share_a}{share_b} \cdot \frac{Supply_b}{Supply_a} \cdot \frac{A}{B}$

According to the lemma, $\frac{A}{Supply_a} = \frac{B}{Supply_b}$

Then, $\frac{Supply_a \cdot A}{Supply_b \cdot B} = 1$

Therefore $\frac{w_a}{w_b} = \frac{share_a}{share_b}$

According to our deposit method, let $d_a \; and \; d_b$ be the deposit amounts, and we have $\frac{d_a}{d_b} = \frac{share_a}{share_b}$

Finally, we know that $\frac{w_a}{w_b} = \frac{d_a}{d_b}$.

11.6 External Exchange Arbitrage

The predictive bond curve relies on arbitrage bots to rebalance the pools. Ideally, pool value can be increased with on-chain arbitrages built into the protocol. Instead of relying on external bots, the DeltaFi protocol performs arbitrages on behalf of liquidity providers. The difficulty is this may lead to higher transaction failure rates, especially considering the congestion on Solana at the time of writing.

11.7 Price Adjustment

To further improve capital efficiency and price impact, $g(m)$ can be modified by keeping the desirable property of pool balancing.

Let $h(m)$ be the modified version of $g(m)$ with the same notations. $h(m)$ is defined as a piecewise linear function:

When $\frac{b}{a} < \frac{B}{A}$, $h(m) = g(m)$.

When $\frac{b}{a} \geq \frac{B}{A}$ and $\frac{b - P \cdot m}{a + m} \geq \frac{B}{A}$, $h(m) = P \cdot m$.

When $\frac{b}{a} \geq \frac{B}{A}$ and $\frac{b - P \cdot m}{a + m} < \frac{B}{A}$, m is split into two parts with $m = m_1 + m_2$.

- m_1 satisfies $\frac{b-P\cdot m_1}{a+m_1} = \frac{B}{A}$.
- Define a more generic curve function $g_0(a,b,m)$ and $g(m) = g_0(a,b,m)$.
- In this case, $h(m) = P \cdot m_1 + g_0(a + m_1, b - P \cdot m_1, m_2)$.

$h(m)$ can also reduce toxic flow from arbitrage activities. It is better implemented together with external exchange arbitrage to mitigate inventory risks.