Pendle Boros AMM

Pendle Finance

August 11, 2025

1 Executive Summary

This white paper outlines the architecture, mechanisms, and features of our AMM system for Boros. Our AMM implements an approach to liquidity provision and ”position” swapping that maintains efficiency and capital optimization.

2 Introduction

AMM is utilized for token swapping between 2 assets. The asset that users trade in Boros is the floating rate generated from an amount of notional token. For example, users can trade the funding rate from openning a long position in ETHUSDT market. This inspires the idea to tokenize the floating stream and create a pool consisting of this token and notional token.

One challenge is that the value of this token is time-sensitive (its value gets smaller as the time for the floating stream gets shorter). We used a variant of the constant product($xy = k$) formula that incorporates time-weighted factor to resolve this problem.

3 Technical Architecture

3.1 Tokenization of interest stream

We define the value of a **float stream token** is the floating rate of 1 notional per annual *from now until maturity*, so its value is:

$$\begin{aligned} 1 \text{ float stream token} &= \text{average floating rate from now to maturity} \\ &\quad \times (\text{time until maturity in years}) \text{ (notional)} \end{aligned}$$

Example: 1 float stream token of a pool when there are 3 months until maturity, given that the funding rate is 10%, has the value of 0.025 notional.

We define the value of a **fix stream token** is 100% of 1 notional per annual *from now until maturity*, so its value is:

$$1 \text{ fix stream token} = (1 \text{ notional}) \times \text{time until maturity in years}$$

Example: 1 fix stream token of a pool when there are 3 months until maturity has the value of 0.25 notional. Note that the value of a fix stream token decays linearly with respect to time until maturity.

Beside the total float stream token the AMM has by opening a long rate position in Boros, we also use an amount of **virtual float stream token**. This amount together with the float stream tokens and fix stream tokens form the **tradable liquidity**.

$$\begin{aligned} \text{tradable liquidity} &= \text{float stream tokens (valued by market rate)} \\ &\quad + \text{virtual float stream tokens (valued by market rate)} \\ &\quad + \text{fix stream tokens (in notional)} \end{aligned}$$

We also let T denote **time until maturity** (in term of years) and t denotes the normalized time until maturity so that $t = 1$ at launch time:

$$t = \frac{T}{\text{total duration}}$$

The AMM operates by its tradable liquidity, meaning it has an underlying amount of float stream tokens and fix stream tokens, represented by its Boros' position and collateral. In order to own 1 float stream token, the AMM needs to open a long position on Boros with notional size of 1 (notional), and this quantity doesn't change overtime without any swap. Meanwhile, any notional from the AMM could be converted into fix stream tokens and vice versa. Assuming that the amount of notional in the AMM is unchanged overtime, the amount of fix stream tokens could be converted will increase linearly, since its value decays linearly. For example, N notional could be converted into $\frac{N}{T}$ fix stream tokens.

Beside tradable liquidity, the AMM also has a buffer to guard it from being liquidated, which would be mentioned in A. Since the AMM holds a position, it has the realized floating stream overtime. This stream goes into buffer, making the pool safer overtime and not affecting tradable liquidity.

3.2 Notation

The quantity of float stream tokens, virtual stream tokens and fix stream tokens are denoted by x, a and y in this white paper. The quantity of buffer is denoted by B (notional) in this paper.

In our contract, the quantity of **total float stream tokens** ($x+a$) is denoted by `totalFloatAmount` and the normalized quantity of fix stream tokens that the AMM owns (in term of notional) in our contract ($y \times t$) is denoted by `normFixedAmount`. These quantity are not affected by the time factor and only changes when there is swap happens. The normalized time until maturity t is often denoted by `timeRatio` in our contract.

3.3 Core Mechanism

Our AMM uses a specialized variant of the shifted constant product formula $(x + a) \times y = k$ that incorporates time-weighted factors and:

$$(x + a)^t \times y = k$$

The purpose of incorporating t in our formula is to make sure that the implied rate of the pool doesn't change over time, given that no swap occured. In fact, the spot price under this formula is:

$$1 \text{ float stream token} = \frac{yt}{x + a} \text{ fix stream token}$$

This implies that the implied APR of the AMM is $\frac{yt}{x+a} \times 100\%$. As we have shown above, this amount doesn't change over time, assuming no swap occured.

3.4 Seeding

At seeding time ($t = 1$), we need to provide the following parameters for the AMM:

- Quantity of floating stream tokens (x): This determines the original exposure to the market of the AMM.
- Quantity of virtual floating stream tokens (a): Note that under our formula, we can let x goes below 0, leading to a short rate position when the implied rate goes higher. This amount of virtual floating stream tokens is the budget for selling floating stream tokens beyond AMM's.
- Market rate (r, where $r = 0.10$ means the market rate is 10%): This determines the implied APR right after we set up the AMM.

- Collateral (C, in term of notional): This is the total amount of notional for fix stream tokens together with the buffer. From 3 parameters above, we could calculate the quantity of fix stream tokens needed:

$$y = (x + a)r$$

  We need to make sure the total collateral is larger than the value of these fix stream tokens:

$$C \geq yT$$

  and the excessive portion becomes the initial buffer:

$$B = C - yT$$

  We also release n LP tokens here for the seeder:

$$L = \sqrt{(x + a)y}$$

  When a user seeds a pool, they follow these steps:

  - Determine the parameters (x, a, r, C) for initial AMM pool as above. Check if $C \geq (x+a)rT$.
  - Open a position of size x with the AMM at market rate r. User holds the short position while the AMM holds the long position and provides $x \times r \times T$ notional to compensate for the obligation of the fix stream of the AMM's position.
  - Provide C notional as collateral for the AMM.

3.5 State of AMM

As mentioned in previous sections, the AMM has a long rate position of size x and an amount of notional. Its underlying state is the tuple $(x+a, yt)$. This tuple, together with other static parameters such as minimum rate m_r, has enough information for implementing the swap logic as shown in 3.6.

3.6 Swap

Note that the swap formula is equivalent to:

$$(x + a)^t yt = k'$$

is constant at a specific time t.

Suppose a user u wants to buy Δ_x float stream token at time t_2 and the current state of the AMM is $(x_1 + a, y_1' t_2)$, last updated at t_1. Note here that the amount of fix stream tokens increased during the duration t_1 to t_2, from y_1 to y_1' such that $y_1' t_2 = y_1 t_1$. The state of AMM is updated as following:

$$\begin{aligned}
k_2' &= (x_1 + a)^{t_2} \times (y_1 t_1) \\
x_2 &= x_1 - dx \\
y_2 t_2 &= \frac{k_2'}{(x_2 + a)^{t_2}} \\
\Delta_y &= \frac{y_1' t_2 - y_2 t_2}{t_2}
\end{aligned}$$

We also need to require:

$$\text{new spot price} = \frac{y_2 t_2}{x_2 + a} \geq m_r$$

The new state is $(x_2 + a, y_2 t_2)$.

Let f denote the swapping fee of the AMM. User u gets Δ_y fix stream tokens which are equivalent to $\Delta_y \times T$ in notional and pays $f \times |\Delta_x|$ fix stream tokens $= f \times \Delta_x \times T$ in notional by collateral as trading fee, which goes to the buffer.

In practice, user and the AMM opens a new swap of size dx (where user longs rate if $dx > 0$ and shorts rate otherwise) with rate $\frac{\Delta_y}{\Delta_x}$ and pays swap fee.

3.6.1 Target swap

Suppose user wants to arbitrage the AMM until target rate r'. The old state is $(x+a, yt)$ and the new state would be $(x'+a, y't)$. Based on 3.3:

$$\begin{aligned}
\frac{y't}{x'+a} &= r' \\
(x+a)^t yt = (x'+a)^t y't &= r' \times (x'+a)^{t+1} \\
x'+a &= \left(\frac{(x+a)^t yt}{r'}\right)^{\frac{1}{t+1}} \\
dx &= x - x'
\end{aligned}$$

User can swap dx out (i.e. open a long position with the AMM of size dx), after that the new rate would be r'.

3.7 Add liquidity

Suppose the number of LP token is L and a user u wants to add liquidity and mint $d \times L$ LP token.

In order to do this action, u has to add $d(B+yT)$ in notional (equivalent to dB into buffer and dy fix stream tokens) and dx float stream tokens (in opening position). Let L_u denotes number of LP tokens that user u has. We need to update the following state change:

$$\begin{aligned}
L_u &= L_u + d \times L \\
L &= L \times (1+d) \\
y &= y \times (1+d) \iff (yt) = (yt) \times (1+d) \\
a &= a \times (1+d) \iff (x+a) = (x+a) \times (1+d)
\end{aligned}$$

The following state changes are updated implicitly:

$$\begin{aligned}
x &= x \times (1+d) \\
B &= B \times (1+d)
\end{aligned}$$

where the former is due to the position provision and the latter is due to the notional provision and the update of y.

3.8 Remove liquidity

Suppose the number of LP token is L and a user u wants to burn $d \times L$ LP token and get liquidity.

After do this action, u has in return $d(B+yT)$ in notional (equivalent to dB in buffer and dy fix stream tokens) and dx float stream tokens. Let L_u denotes number of LP tokens that user u has. We need to update the following state change::

$$\begin{aligned}
L_u &= L_u - d \times L \\
L &= L \times (1-d) \\
y &= y \times (1-d) \iff (yt) = (yt) \times (1-d) \\
a &= a \times (1-d) \iff (x+a) = (x+a) \times (1-d)
\end{aligned}$$

The following state changes are updated imlicitly:

$$\begin{aligned}
x &= x \times (1-d) \\
B &= B \times (1-d)
\end{aligned}$$

where the former is due to the position withdrawal and the latter is due to the notional withdrawal and the update of y.

4 Negative AMM

When the rate of the AMM is negative, the value of float stream token is negative. Therefore we define **flipped float stream token**:

$$\begin{aligned} 1 \text{ flipped float stream token} &= -1 \text{ float stream token} \\ &= -\text{average floating rate from now to maturity} \\ &\times (\text{time until maturity in years}) \ (\text{notional}) \end{aligned}$$

Maintaining all other parameters constant, we modify the pool to hold flipped float stream tokens instead of standard float stream tokens. This results in a short rate position of magnitude $|x|$. Since all algorithms now operate on flipped float stream tokens, we convert the inputs and outputs of notional size into float stream tokens using the mapping $x \mapsto -x$.

Appendices

A Liquidation price

From the AMM's formula, one can see that as implied APR goes to 0, the total size (x) of the AMM goes to infinity, and could make the AMM becomes liquidatable. To prevent this effect, we make a requirement of minimum rate m_r, which limit the maximum size of the AMM, and hence guard it from being liquidated. In this section, we study the rationale of choosing this parameter.

Firstly, we ignore the minimum rate requirement. As the implied rate decreases, the size of the AMM increases, so the maintainance margin also increases while the total value of the AMM decreases. At some point, the `marginRatio` of the AMM would decrease and reach 1, making it liquidatable. We call the implied rate at that point **unconstrained liquidation price**, denoted by r_0.

Obviously, we need to set $m_r > r_0$. As the exposure of the AMM at m_r is lower than at r_0, the true liquidation price would also be lower than r_0. Therefore, increasing the minimum rate, m_r reduces the AMM's exposure at that point and lowers the liquidation price.

The following is the calculation for r_0.

At the state where the rate is r_0:

$$x + a = \frac{kt^{\frac{1}{t+1}}}{r_0^{\frac{1}{t+1}}}$$

$$y = \frac{r_0(x+a)}{t} = \frac{kr_0^{\frac{t}{t+1}}}{t^{\frac{t}{t+1}}}$$

The total value is:

$$B + (y + xr_0)T$$

We let MMR denotes Maintainance Margin Ratio. Since the `marginRatio` is 1 in this state:

$$\begin{aligned} B + (y + xr_0)T = x \times T \times MMR \\ \frac{B}{T} + \frac{kr_0^{\frac{t}{t+1}}}{t^{\frac{t}{t+1}}} + \left(\frac{kt^{\frac{1}{t+1}}}{r_0^{\frac{1}{t+1}}} - a\right)(r_0 - MMR) = 0 \end{aligned}$$

This equation has unique solution for r_0 as the left hand side is increasing w.r.t r_0. We can find it numerically.

Now if we enforce a min tradable trade on the AMM $m_r > r_0$, then the true liquidation price would be l:

$$\begin{aligned} B + (y + xl)T = x \times T \times MMR \\ \iff l = MMR - \frac{B/T + y}{x} \end{aligned}$$

References