CBMM Pool: A Constant Burn Market Maker

Your Name
Your Affiliation
`your.email@example.com`

October 31, 2025

Abstract

This whitepaper describes the CBMM (Constant Burn Market Maker) pool, an improved version of the standard constant product market maker (CPMM) that automates token burns and buybacks resulting in a positive impact on the token's price and more continuous demand.

Contents

1 Introduction

- CPMM, CLMM - originally pioneered by Uniswap? - mention Pump.fun - virtual reserve - define insolvency - price impact of burn - burn structure (in instructions) - price impact of the automated buybacks

2 Background

- The goal is to design a mechanism that allows us to reward off-chain behavior by creating a direct on-chain impact on the beans price by burning the beans' supply. - By burning the supply we are lowering liquidity, creating scarcity. - We need to make sure that even after burn the pool is able to withstand sale of all the tokens even if that means that some users might be selling for a price that is worse than the starting one. - We have two mechanisms that help us with the positive price action on-chain - burning and continuous conditional buybacks. These mechanisms are very closely intertwined and complement each other. - The goal is to create a mechanism that rewards off-chain actions long term and by inducing the positive price action it motivates people to on-chain actions and speculation. - There is no free lunch, if we burn tokens "someone" has to pay for it. - two possible approaches - burn of a percentage of the pool supply/ burn of the total supply percentage - motivation for the virtual reserve is that otherwise it would be very cheap to snipe a big portion of the reserve. The virtual reserve helps to mitigate this by effectively setting a starting price for the token. If no burning is involved, the token price in this pool can never drop under this price. However, if burning is involved, the needed virtual price drop lowers the minimal price that the token can drop to and thereby keeps the pool healthy. - A slow transition from a purely virtual scaling system into a progressively collateralized one. - THis is a token rollout mechanism, it is not designed for tokens that are already in the market.

2.1 Conventional Market Making Strategies

- Description of standard CPMM pool, note about CLMM pools, DAMM pools - Short description of bonding curves - Neither of these has any mechanisms for off-chain action motivation except for a manual buyback and burn. We are replacing this with an automated solution

3 Mathematical Model

We first define the key terms used throughout this section.

Definition 3.1. The **Virtual Token Reserve** V is the portion of the token reserve that is not backed by actual assets.

Definition 3.2. The **Real Token Reserve** A is the token reserve in the pool that is backed by actual assets.

Definition 3.3. Insolvency is the state of the pool where its reserves are insufficient to accommodate the sale of all outstanding beans.

We assume the initial Real Token Reserve reserve is zero. CBMM is designed as a token rollout mechanism, so we do not need to account for any existing supply outside the pool. The virtual reserve sets the initial price of the token.

3.1 Trading

Buys and sells in CBMM follow the mechanics of the standard CPMM with a virtual reserve; we include them here for completeness. Denote the pre-trade (buy or sell) values as:

$$\begin{aligned}
A_0 &= A,\\
B_0 &= B,\\
V_0 &= V,\\
k_0 &= k = (A+V)B
\end{aligned}$$

During trading, the invariant k and the virtual reserve V do not change. The amount of beans b received by the user when spending ΔA tokens (which increases the pool reserve to $A + \Delta A$) follows from

$$k = (A + \Delta A + V)(B - b), \tag{1}$$

which gives:

$$b = B - \frac{k}{A + \Delta A + V}. \tag{2}$$

Similarly, the amount of tokens a received by the user when spending ΔB beans (which increases the pool reserve to $B + \Delta B$) follows from

$$k = (A - a + V)(B + \Delta B), \tag{3}$$

which gives:

$$a = A + V - \frac{k}{B + \Delta B}. \tag{4}$$

The current price of beans is:

$$P = \frac{A+V}{B}. \tag{5}$$

3.2 Beans Supply Burning

Let the initial state of the pool be:

$$\begin{aligned}
A_0 &= 0,\\
B_0 &= B,\\
V_0 &= V,\\
k_0 &= (0+V)B = VB
\end{aligned}$$

Assume, without loss of generality, that trading occurs before the burn, lowering the beans reserve by $x \geq 0$. The post-trade values are then:

$$\begin{aligned}
A_1 &= \frac{Vx}{B-x},\\
B_1 &= B - x,\\
V_1 &= V,\\
k_1 &= k_0
\end{aligned}$$

Now, if we burn y beans with $0 \leq y < B - x$, the post-burn state is:

$$\begin{aligned}
&A_2 = \frac{Vx}{B-x}, \\
&B_2 = B - x - y, \\
&V_2 \text{ to be determined}, \\
&k_2 = (A_2 + V_2)B_2
\end{aligned}$$

To ensure the pool is not insolvent, if everyone sells their beans back to the pool, there must be sufficient tokens to satisfy the sale. The post-sale state must satisfy:

$$\begin{aligned}
A_3 &\geq 0, \\
B_3 &= B - y, \\
V_3 &= V_2, \\
k_3 &= (A_3 + V_3)B_3
\end{aligned}$$

From $k_2 = k_3$ and $k_3 \geq V_3 B_3$, we obtain a bound on V_2 that ensures the pool is not insolvent:

$$V_2 \leq \frac{V(B-x-y)}{B-x}. \tag{6}$$

Thus, after every burn, the virtual reserve must be adjusted downward to avoid insolvency. A natural choice is to set V_2 to the maximum value satisfying the bound, which minimizes price impact.

4 Implementation

- We set A = 0

5 Analysis

- burns themselves don't pump the price (by much?) - volume itself doesn't pump the price - these two mechanisms complement each other and motivate users to do one or another. - real initial reserve is useless as it is immediately used

6 Conclusion

- We are rolling this out as a closed system with beans, but the concept is applicable to any asset pair - Could be generalized, V could go under 0 to be able to keep the pool solvent if starting token reserve is greater than zero and we want to retain it after all the beans return to the pool.