

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	2
2	Background	2
2.1	Conventional Market Making Strategies	2
3	Mathematical Model	2
3.1	Trading	3
3.1.1	Beans Supply Burning	3
4	Implementation	4
5	Analysis	4
6	Conclusion	4

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

Let's first define a bunch of terms that we will use in the mathematical model.

Definition 3.1. The **virtual reserve** V is the reserve of the pool that is not backed by any actual assets. It is used to set the starting price for the token.

Definition 3.2. The **real reserve** A is the reserve of the token in the pool that is backed by actual assets.

Definition 3.3. **Insolvency** is the state of the pool where the reserve of the pool is not enough to cover the sale of all the tokens.

We will assume that the initial reserve of the token is 0 as the CBMM is designed as a token rollout mechanism and therefore we do not need any initial token reserve to account for any existing token supply. Moreover, we can use the virtual reserve to set the starting price for the token.

3.1 Trading

The concepts of buys and sells in CBMM pool are the same as in the standard CPMM pool with a virtual reserve and are included for completeness. Let's denote the values before the trade (buy or sell) as follows:

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

It is important to note that during trading the pseudoconstant k and virtual reserve V do not change and, therefore, we can derive the amount of beans Δ_B received when spending Δ_A tokens by solving the equation $k = (A + \Delta_A + V)(B + \Delta_B)$ for Δ_B :

$$\Delta_B = \frac{k}{A + \Delta_A + V} - B \quad (1)$$

and the amount of tokens Δ_A received when spending Δ_B beans by solving the equation for Δ_A :

$$\Delta_A = \frac{k}{B + \Delta_B + V} - A \quad (2)$$

The current price of the beans is given by:

$$P = \frac{A + V}{B} \quad (3)$$

3.1.1 Beans Supply Burning

Let the initial state of the pool be as follows:

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

Let's assume without loss of generality that there is some trading happening before the burn which lowers the balance of beans reserve by $x \geq 0$. Then the values after the trading will be:

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

Now if we burn y such that $0 \leq y < m - x$ from the supply of beans, the values after the burn will be:

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

And finally to ensure that the pool is not insolvent we to ensure that if everyone sells their beans back to the pool we have enough tokens to facilitate this. Therefore, we need to ensure that the state after the sale is as follows:

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

From the equations $k_2 = k_3$ and $k_3 \geq V_3 B_3$ we can derive the following inequality for the value of V_2 for the pool to not be insolvent:

$$V_2 \leq \frac{V(B-x-y)}{B-x} \quad (4)$$

This concludes that after every burn the virtual reserve must adjusted downwards to ensure that the pool is not insolvent. It is reasonable to minimize the adjustment by setting V_2 to the maximum value that satisfies the inequality as this will result in the smallest possible 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.