CBMM Pool: A Constant Burn Market Maker

Your Name
Your Affiliation
your.email@example.com

November 5, 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

- CFMM, CPMM, CLMM, DAMM etc.
- mention Pump.fun, Uniswap
- 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.
- Points (beans) vs tokens

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

In this section we will define the mathematical model of the CBMM pool. We only focus on the token trading and burning mechanics, the buyback mechanics are an extension of this model and therefore will be discussed in the **??** section.

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.

3.2.1 Price impact of the burn

Denote the price before burn as $P_1 = \frac{A_1+V_1}{B_1}$ and the price after burn as $P_2 = \frac{A_2+V_2}{B_2}$. Substituting the values from the previous section, where $B_1 = B - x$ and $V_2 = \frac{V(B-x-y)}{B-x}$, we obtain:

$$P_1 = \frac{A_1 + V_1}{B_1} = \frac{\frac{Vx}{B-x} + V}{B - x} = \frac{VB}{(B-x)^2},$$
$$P_2 = \frac{A_2 + V_2}{B_2} = \frac{\frac{Vx}{B-x} + \frac{V(B-x-y)}{B-x}}{B - x - y} = \frac{V(B-y)}{(B-x)(B-x-y)}.$$

The relative price impact of the burn is then:

$$\frac{P_2 - P_1}{P_1} = \frac{xy}{B(B-x-y)}. \tag{7}$$

This formula shows that the relative price impact is proportional to the product xy of beans bought (x) and beans burned (y), divided by $B(B-x-y)$. The impact increases with the amount of beans held outside the pool (x), meaning burns have greater price impact when more tokens have been purchased. If $x = 0$ (no tokens purchased), the burn has no price impact, as expected.

3.3 Other invariant types

Definition 3.4 (Power CBMM)**. Power CBMM** generalizes the standard CBMM by using a power parameter $p > 0$ in the invariant:

$$k = (A + V)B^p, \tag{8}$$

where A is the token reserve, B the bean supply, V the virtual reserve, and p is a real parameter controlling the curve's shape.

When $p = 1$, this reduces to the standard CBMM invariant. The parameter p controls the curvature of the bonding curve: $p > 1$ produces a steeper curve, while $0 < p < 1$ produces a flatter curve. This allows fine-tuning of the pool's behavior, particularly affecting initial token purchases and the distribution of tokens to early adopters.

The trading formulas for Power CBMM follow from the invariant $k = (A + V)B^p$. When spending ΔA tokens to buy beans, the amount of beans b received is:

$$b = B - \left(\frac{(A+V)B^p}{A + \Delta A + V}\right)^{1/p}. \tag{9}$$

When spending ΔB beans to sell, the amount of tokens a received is:

$$a = (A + V)\left(1 - \frac{B^p}{(B + \Delta B)^p}\right). \tag{10}$$

After burning y beans when x beans have already been purchased, the virtual reserve must be adjusted to maintain solvency. The adjusted virtual reserve V_2 is:

$$V_2 = \frac{V\left((B-x)^p - B^p\right)(B-x-y)^p}{(B-x)^p\left((B-x-y)^p - (B-y)^p\right)}, \tag{11}$$

where $(B-x)^p\left((B-y)^p - (B-x-y)^p\right) \neq 0$.

We have considered other invariant types as Weighted geometric mean but it shows that the virtual reserve formulas are too complex to be conveniently implemented in the on-chain program using integer arithmetic.

4 Implementation

This section describes an on-chain implementation of CBMM as a Solana program. The principles described are specific to CBMM, otherwise the implementation uses the same approach as standard CPMM pools. We first present the Continuous Conditional Buybacks mechanism, which partially offsets the required reduction in the virtual reserve V after burns. We then outline safety controls for burn operations and note key configuration considerations. The design objective is to minimize user friction while faithfully realizing the mathematical model from §3.

4.1 Continuous Conditional Buybacks

Burns require a reduction of the virtual reserve from V_1 to V_2 (with $V_2 \leq V_1$). This adjustment reduces the positive impact of the burn. We implement *Continuous Conditional Buybacks* (CCB) to partially offset this by redirecting a portion of trading fees into the real token reserve A, which bumps the price slightly back up.

Let $\Delta V := V_1 - V_2 \geq 0$ denote the required reduction in virtual reserve implied by §3. The mechanism is:

- **Fee accumulation**: On each trade, a fixed fraction of token fees is routed to a dedicated on-chain fee vault (an associated token account controlled by the program).
- **Burn-time top-up**: Upon a burn event, the program deposits into A an amount $\Delta A_{\text{topup}} := \min(\Delta V,\ F)$, where F is the available collected fee balance. Any shortfall $L := \Delta V - \Delta A_{\text{topup}} \geq 0$ is stored in pool state as an outstanding liability.
- **Continuous repayment**: While $L > 0$, subsequent trades continue to contribute fees that are immediately applied to reduce L until it reaches zero. Any overage remains in the fee vault and is handled by the fee accumulation rule.

The choice of top-up target $V_2 - V_1$ is an arbitrary decision, other targets are possible. We use it because it naturally "compensates" for the virtual reserve reduction, even though this compensation is not exact as the amount in the real token reserve can be later extracted while trading. Technically, this is not a buyback, as no beans leave the pool. However, adding tokens to A increases the beans' price and effectively buys back part of the burn's price impact.

4.2 Burn Safety Mechanisms

To bound operational risk and align burns with off-chain incentives, we introduce two optional controls: (i) per-user daily limits and (ii) an authorization gate for burns.

4.2.1 Daily burn limits

We introduce a lightweight on-chain meter, `UserBurnAllowance`, keyed by user. It records the number of burns over a rolling 24-hour window and the timestamp of the most recent burn. Account creation is permissionless (similarly to SPL Associated Token Accounts) so any party can create and fund an account to bootstrap a user's allowance. To avoid rent leakage, accounts associated with users inactive for ≥ 24 hours may be closed and rent returned to the original funding wallet. If a user exceeds the configured daily limit, additional burn attempts are rejected until the window resets.

For lowest-cost deployments, equivalent functionality could be realized via State Compression (e.g., Merkle-tree-based indexing), at the cost of off-chain infrastructure dependence.

4.2.2 Burn authority

On every burn we require a signature from a Burn Authority. This ensures that all burns are tied to specific off-chain activities and that users have not simply automated the burn operation by calling the program directly. This mechanism is optional and can be turned off if decentralization is a priority.

4.3 Pool Configuration Considerations

The main decision that influences the pool behaviour is the initial virtual reserve V. This reserve directly sets the token initial price and is linearly related to the trading volume needed to be able to top-up the pool fully. Moreover, the higher initial reserve the harder it is to snipe a large amount of beans at the starting price.

To be able to mitigate the sniping issue we can not only use higher virtual reserve, but also use Power CBMM with $p > 1$ to increase the price impact of the initial purchases. However, with higher p the on-chain program needs to use more complex arithmetic to implement the trading formulas and ensure that there is no overflows or significant precision loss that would affect the pool's health. From our tests we have found that $p = 2$ is a good compromise between complexity and effectiveness.

5 Conclusion

This work set out to design a market-making mechanism that can translate verifiable off-chain activity into on-chain price impact through controlled burns, while preserving the pool's solvency. We constructed a mathematical model for CBMM with a virtual reserve V and derived closed-form expressions for trading, burning, and the induced price impact. On the implementation side, we outlined a Solana program architecture that realizes these mechanics, introduces Continuous Conditional Buybacks (CCB) to partially offset the virtual-reserve reduction after burns, and adds safety controls to regulate burn frequency and authorization.

The model reveals a clear separation of roles. First, burns directly increase price with a relative impact proportional to the amount of beans outside the pool and the burned amount. The pool trading volume by itself does not directly contribute to the price increase but complements burns by compensating for the virtual reserve detoriation through the Continuous Conditional Buybacks mechanism. Because burning reduces the in-pool supply, it increases price but also increases volatility. The virtual reserve acts here as a bridge mechanism to facilitate a slow transition from a purely virtual scaling system into a progressively collateralized one. In healthy conditions, as V is reduced toward its minimum, the pool matures by price becoming increasingly supported by real collateral rather than virtual scale.

From a design perspective, the initial virtual reserve V sets the starting price and determines how much trading volume is required to fully top up the pool via fees. Larger V makes early sniping more costly. The optional Power CBMM variant controls curve shape via a parameter $p > 0$; choosing $p > 1$ concentrates price impact at earlier purchases and can be used to mitigate initial sniping risk, while $0 < p < 1$ flattens the curve.

This paper describes the CBMM pool as a token rollout mechanism, but generalizing CBMM to other asset pairs is feasible. Possible extensions and enhancements include allowing third-party liquidity provision (LP) in a controlled manner and analyzing CBMM pool with nonzero initial token reserve that should be preserved by the burn mechanism. These mechanisms might require $V < 0$ and is left for future study.