Doppler Multicurve

April 2025

Austin Adams
austin@whetstone.cc

Matt Czernik
matt@whetstone.cc

Rohan Kulkarni
rohan@zora.co

Cooper Kunz
cooper@whetstone.cc

Abstract

Financial markets are increasingly becoming internet native. One-size-fits-all token launchpads and liquidity bootstrapping protocols have failed to meet the needs of token creators and their communities. Developers increasingly want onchain markets with application-specific customizations. We introduce Doppler Multicurve, a new primitive for onchain issuance that enables multiple price curves to be specified during an initial auction or liquidity bootstrapping phase, which can then be tailored to a specific use case or application's needs. In doing so, developers have the expressivity necessary to reduce maximal extractable value (MEV), and enable more capital-efficient liquidity formation.

1 Introduction

When utilizing a concentrated liquidity automated market maker (CLAMM), static bonding curve-based projects place one constant liquidity position for buyers. However, one constant liquidity placement comes with a subtle but impactful downside: significantly more tokens are available for trading at the cheapest prices on the position, leading to an outsized portion of the token supply sold to the earliest buyers (who are generally MEV bots). MEV bots, in turn, quickly dump these cheap shares when new buyers enter the market, extracting value from both the new buyers and the token project. Additionally, the wider the position across the price spectrum, the more outsized this price dilation effect is, and potential downside for users.

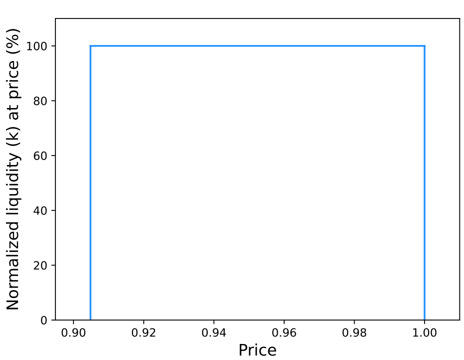


Figure 1: A Single Constant Liquidity Position

Doppler Multicurve turns this on its head and addresses one of the biggest limitations of liquidity-bootstrapping protocol's integration with CLAMMs. Instead of placing one constant liquidity position (shown in Figure 1), integrators specify their desired curves, which in itself is entirely managed by Multicurve. Multicurve is entirely compatible with the current Doppler Protocol [1] ecosystem.

In practice, this means that Multicurve creates a curve mimicking a log-normal curve utilizing individual concentrated liquidity positions, which aims to sell a constant number of tokens in each price bucket. An example of this is shown above in Figure 1. Additionally, while not shown in the above figure, multiple of these curves can be placed, allowing customization of complex curves and price paths, entirely maintained and executed inside the existing Doppler position manager. Similar to all other Doppler features, it's natively compatible with existing popular AMM integrations, such as Uniswap.

Compared to a static bonding curve, Multicurve reduces sniping, leads to higher liquidity and prices, allows for a more customizable market structure, and lowers the cost of price discovery.

2 Background

Over the last year, there has been a significant increase in the number of bonding curve-based liquidity bootstrapping protocols such as pumpdotfun, clanker, and Meteora. This proliferation has resulted in a awide variety of different implementations with various market structure impacts. The bonding curve design utilized by most (with notable exceptions like pumpdotfun) utilizes an $x \cdot y = k$ concentrated-liquidity AMM (CLAMM), such as Uniswap v3 [2], as the matching engine for their protocols.

However, as previously stated, $x \cdot y = k$ AMMs have a major drawback for new token liquidity bootstrapping, where the amount of tokens sold by the protocol decreases as the price of the token goes up, which is guaranteed if there is any trading activity. By selling the most amount of tokens at the cheapest possible price, token projects that use these static bonding curves will have increased losses from sniping, increased cost of price discovery, and will result in overall lower user welfare. Additionally, we will show mathematically in the appendix that selling more tokens during the earlier trading period phase is a fundamental property of $x \cdot y = k$ AMMs, which is addressed through a layering technique pioneered by Doppler Multicurve.

3 Single Curve

At its core, DOPPLER MULTICURVE is first a collection of individual single log-normal curves. With multiple curves, integrators can describe increasingly complex curvatures by combining multiple single curves.

First, we describe the parameterization of a single curve. We note for brevity that we only describe the case where the new token is $token_0$.

To parameterize a single curve, integrators provide a $tickLower$, $tickUpper$, total token amount ($amount$) for bootstrapping liquidity, and desired steepness (known as $numPositions$). The $numPositions$ is the number of positions created by MULTICURVE.

$tickLower$ and $tickUpper$ refer to the placement of possible liquidity positions in CLAMMs and must satisfy the conditions enforced by the specific AMM.

To start, the total token amount, $amount$ is divided equally among all positions.

$$positionTokens_i = \frac{amount}{numPositions} \tag{3.1}$$

Next, in the $token_0$ case, each position ends at $tickUpper$ with the $tickLower$ of the the position linearly spanning from $tickLower$ to $tickUpper$. This is flipped accordingly in the $token_1$ case.

$$tickLower_i = tickLower + \lfloor \frac{i \cdot (tickUpper - tickLower)}{numPositions} \rfloor \tag{3.2}$$

Due to both integer math and implementation details, each $tickLower_i$ is binned according to the $tickSpacing$. Depending on the configuration of the curve and the $numPositions$, this could result in multiple positions overlapping with the same $tickLower$ and $tickUpper$. This should be avoided due to unnecessary gas costs, but is accounted for in the implementation.

Each position is placed from its calculated $tickLower_i$ to $tickUpper$. Each of these positions are given equal numbers of tokens with their $liquidity$ calculated at run time, resulting in a staircase-like liquidity structure, where positions closer to $tickUpper$ have more liquidity. This results in more tokens concentrated around that value than a constant liquidity position.

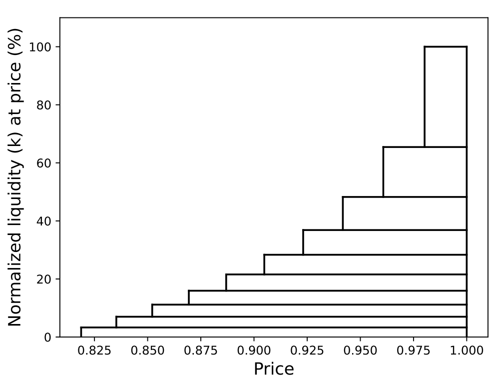


Figure 2: A selected single curve

In Figure 2, we show a sample implementation of the single curve design with $tickLower = -2000$, $tickUpper = 0$, $numPositions = 10^1$. With more positions, the $liquidity$ of the system is more concentrated around global $tickUpper$ of the curve. More liquidity in the upper parts of the curve results in faster price accumulation and a higher average price execution.

From these four parameters, every position in the curve can always be cheaply calculated ensuring that all positions are always tracked while additionally lowering the gas cost by maintaining less information inside onchain storage.

4 Multicurve

In principle, MULTICURVE is a direct extension of a single curve, allowing integrators to provide multiple curve parameterizations that are then jointly managed by the DOPPLER position manager. By allowing integrators to place multiple curves, increasingly complex curvatures can be described, resulting in the ability to place bonding curves that are molded around the design of the token and the expected price path. As different token types will have different expected price paths, features, and timelines, the structure of the liquidity bootstrapping should also adapt to these features as well.

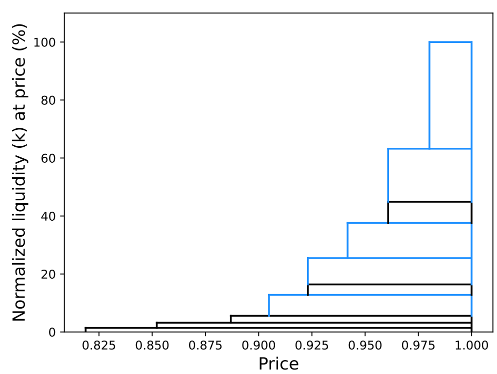


Figure 3: Two selected multiple curvitures

An example MULTICURVE liquidity path is shown above. The two curves are $Curve_1$ colored in black with ($tickLower_1 = -2000$, $tickUpper_1 = 0$, $numPositions_1 = 5$, $tokenAmt_1 = 10\%$) and $Curve_2$ colored in blue with ($tickLower_2 = -1000$, $tickUpper_1 = 0$, $numPositions_1 = 5$, $tokenAmt_1 = 25\%$). This curve is designed to facilitate early price discovery from $Curve_1$ with thin but continuous liquidity. From there, a $Curve_2$ is layered over the first to lower price impact at higher prices. Both of these curves are entirely managed by the DOPPLER position manager

As an added feature, DOPPLER itself calculates the amount of the numeraire token (generally ETH, but there is no requirement) that will result in exiting the entire curve. This allows integrators to easily know how many tokens will be generated during initial liquidity bootstrapping. This also allows DOPPLER itself to calculate the optimal amount of the quote token (the token being sold)

[1] As $liquidity$ is normalized in the chart, $amount$ is arbitrary

needed to bond. Finally, this ensures that Doppler has enough of the bootstrapping token liquidity to bond against as a safety check.

5 Conclusion

As more of the financial economy moves onchain, and internet capital markets become increasingly relevant, it is important to have practical mechanisms that can support their ambitions. The design of Doppler Multicurve is intended to simplify the integrator experience while allowing them to customize their capital markets to their application or implementation needs. Integrators pick what their starting and ending price of the token is, and then go end to end with cheaper price discovery.

For one example, by utilizing just one single curve in Multicurve, Zora lowered the starting price of their tokens by 60x (from $1,320 to $22) with the same cost from sniping as a constant liquidity curve.

By lowering the cost of sniping with high price precision, the tokens created on the protocol will result in higher prices and more liquidity, as less value will be extracted via sniping bots. By incorporating a second curve and iterating on the design according to market feedback, the cost of initial liquidity provision should continue to lower.

A Mathematical Appendix

To explain why a constant liquidity position leads to more tokens sold at cheaper prices, we first must explain what constant liquidity in concentrated liquidity math refers to.

A good mental model is that *liquidity* is the k in the ever popular $x \cdot y = k$ equation.[2]

First, we define the price p as the reserves of x divided by the reserves of y.

$$p = \frac{x}{y} \tag{A.1}$$

Next, we can rewrite and substitute x to find the impact of *price* increasing.

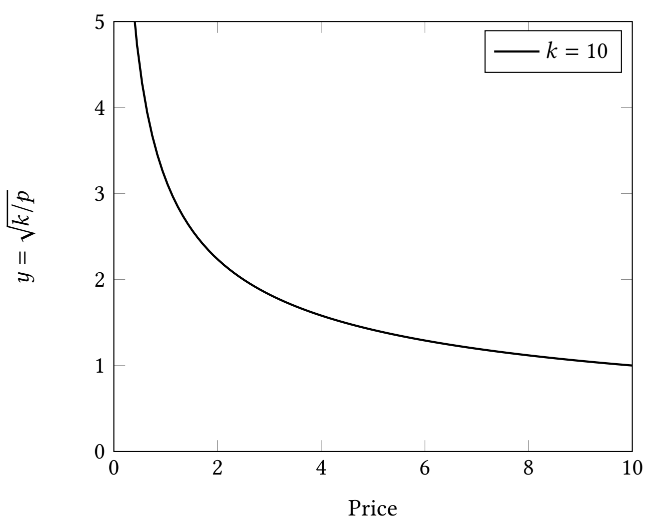
$$k = y^2 p \tag{A.2}$$

Finally, we can rewrite to isolate y.

$$y = \sqrt{\frac{k}{p}} \tag{A.3}$$

Since $p = \frac{x}{y}$, an increase in *price* means that y is more valuable than it previously was worth. We see that as the price increases, the amount of y in the pool likewise must decrease. This is fairly obvious because users are buying y with x if the price is going up, resulting in more x and less y in the pool.

Plot of $y = \sqrt{k/p}$ with $k = 10$



However, what is important to note is actually the shape of this graph. It is non-linear, meaning that the amount of tokens sold between two bins is not constant.

Now, we want to show that there always exists more tokens in earlier price bins than in later price bins if k is constant in the bin. This results in the amount of y sold to be larger at the earliest stages inside the liquidity pool. This effect results in an outsized portion of the token supply sold to the earliest buyers.

To move between arbitrary prices in concentrated liquidity, we must buy all the tokens y from p to $p + \delta$.

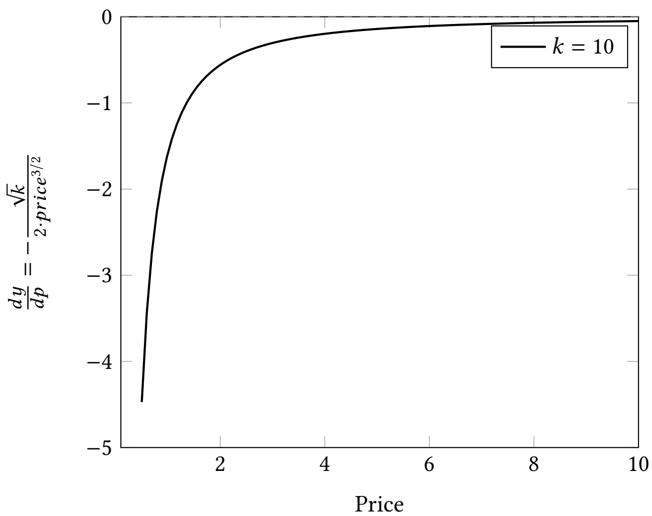
[2] We note that in the Uniswap v3 model, liquidity is actually sqrt(k) due to implementation details, but this does not notably impact the directionality of the meaning.

To show that the amount y is strictly decreasing from p_1 to p_2 if $p_2 > p_1$, we must show that the derivative of y with respect to $price$ is always negative.

$$\frac{dy}{dp} = -\frac{\sqrt{k}}{2p^{3/2}} \tag{A.4}$$

This derivative is negative for all positive values of p and k, which means the function is always decreasing as price increases. This results in more tokens sold at the earliest bins than later ones.

Derivative of $y = \sqrt{k/p}$ with respect to p



Additionally, the derivative is non-linear and increases rapidly around the earliest possible prices, resulting in this effect to be larger at the start of the price curve and for smaller token prices (which is what is commonly used for new token projects)

Overall, the main way to combat this effect is to utilize a shifting k value, which is all implemented via Doppler Multicurve. As shifting the amount of k between bins negates this non-linear decrease in the amount sold per bin.

References

[1] Austin Adams, Matt Czernik, Clement Lakhal, and Kaden Zipfel. 2025. *Doppler: A liquidity bootstrapping ecosystem.* Retrieved Apr 21, 2025 from https://github.com/whetstoneresearch/docs/blob/main/whitepapers/doppler/Dutch_auction_Dynamic_Bonding_Curves.pdf

[2] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. 2021. *Uniswap v3 Core.* Retrieved Jun 12, 2023 from https://uniswap.org/whitepaper-v3.pdf

B Disclaimers

No Legal, Financial, or Investment Advice. This document does not constitute legal advice, financial advice, investment advice, trading advice, or a recommendation of any kind by Doppler, its affiliates, or any of their respective officers, directors, managers, employees, agents, advisors, or consultants. No information contained herein should be relied upon as the basis for any investment decision, contract, or any other decision regarding engagement with Doppler or any of its projects.

Forward-Looking Statements. This document may contain forward-looking statements based on assumptions and beliefs of Doppler, its officers, or its affiliates. These statements are subject to known and unknown risks, uncertainties, and other factors, many of which are outside Doppler's control, that may cause actual outcomes to differ materially from the projections or expectations set forth in such statements. Doppler makes no obligation to update or revise any forward-looking statements to reflect subsequent events, developments, or changes in circumstances after the date on which such statements were made, except as required by law.