λ - Constant Function Markets Generalizing and Mixing Automated Market Makers

Giorgos Felekis
Advanced Blockchain Research
giorgos@advancedblockchain.com

Jesper Kristensen
Advanced Blockchain Research
jesper@advancedblockchain.com

***Abstract*—One of the most exciting recent developments in Decentralized Finance (DeFi) has been the development of decentralized exchanges, called Automated Market Makers (AMMs). In this work, we study the most prominent special class of them, the Constant Function Market Makers (CFMMs). We introduce a generalized formula for CFMMs, called λCFMMs, which encapsulates the idea of combining the advantages of constant sum and constant mean CFMMs by blending their functions where λ is the degree of mixture. Our experiments demonstrate the behaviour of this generalized formula for various token pools with different properties and price differences, and evaluate its performance regarding slippage and impermanent loss for different degrees of mixture during a trading period. We further show that given the nature of the pool and an optimization objective, different levels of mixture lead to optimal non-trivial functions, which as we show, outperform some of the most popular AMMs such as Uniswap. The novelty of λCFMMs is both the mixing method that helps us target more efficient AMM functions and also the fact that motivates the idea of dynamic AMM functions that given certain features can self-adjust their parameters in order to produce mutual profits for both the traders and the liquidity providers.**

1. Introduction

Market makers in traditional finance (TradFi) such as the ones for stocks, offer liquidity for enabling investors to sell or purchase assets that are close to a publicly listed price. In TradFi the mechanism of order books is the one determining the prices of different assets via a trusted third party who keep records of all the bids and asks.

However, this approach changed with the introduction of decentralized finance (DeFi). Blockchains have now a new purpose of offering financial services without the need of a middleman. This need introduced the idea of Decentralized Exchanges (DEXes), which allow users to trade in a fully noncustodial manner. The most recent blockchains are heavily dependent on the concept of smart contracts (SCs). SCs are program objects that live on the blockchain and they are able to communicate with one another, via message-calls within the same execution context and support atomicity, i.e.the concept that a transaction either success or fails entirely. In such a context the order book mechanism will be challenged to properly operate mainly due to the expensive nature of computation and storage that they demand. A reasonable approach in order to avoid these costly on-chain manipulations was to implement trades off-chain [2] [3]. Although the gas paid to the miners in such a case decreases there are still certain issues that these suffer from, mainly regarding front-running opportunities [4] that lurk behind.

Lately, all of the above motivated a new DEX, called curve-based Automated Market Makers (AMMs). The study of AMMs initially began from the field of algorithmic game theory and the study of scoring rules within the statistics literature [6]. The first to propose a pricing mechanism based on these scoring rules was Robin Hanson [7], who introduced the Logarithmic Market Scoring Rule (LMSR). In contrast to AMMs, these early automated market makers proved to be computationally expensive and cumbersome for the users. Recent AMM exchanges such as Bancor [5], Uniswap [8] [11], StableSwap/Curve [9] have become undeniably popular, something that is also proven by the fact that the trading volume on all these DEXes exceeded $60 billion in March 2022.

The idea of AMMs is based on liquidity pools where third parties pool their assets into reserves and then based on a pre-defined mathematical formula (curve), the asset price is been determined. In these DEXs, liquidity providers deposit amounts (of equal value) of multiple types of assets to the designated liquidity pools, and traders exchange against the pools of tokens instead of relying on order matching. Curve-based AMMs provide a continuous supply of liquidity

978-1-6654-6104-7/22/$31.00 ©2022 IEEE
DOI 10.1109/Blockchain55522.2022.00047

compared to the order book model and depending on a pre-defined mathematical function (curve), they can potentially allow for a wide range of exchange prices.

1.1. Contributions

In short, the main contributions of our work are:

- The introduction of a generalized AMM formula that can interpolate between Constant Sum and Constant Mean functions while containing both as special cases.
- The analysis of the theoretical properties of this new family of functions and the proof of various properties regarding the pricing and the trading functions of it.
- The extensive experimentation on real-world trading periods accompanied with novel visual representations regarding the tracking of slippage and impermanent loss behaviour for different mixture coefficients.
- Insights that indicate the superiority of our formula in targeting more efficient "intermediate" functions compared to the standard markets by adjusting the mixture level.

2. Background

2.1. Constant Function Market Makers

Constant Function Market makers (CFMMs) constitute the most popular class of AMMs and the first to be applied in real world decentralized applications. Their main purpose is to provide decentralized exchanges of digital assets and are based on a function that establishes a pre-defined set of prices based on the available quantities of two or more assets. Their main difference with traditional market makers and order book based exchanges is the fact that the trading happens against a pool of assets rather than a specified third party. One can think CFMMs as a "game" between two players and a group of supervisors. The two players are the *Traders*, who exchange assets on the pool, and the *Liquidity Providers (LPs)*, who are willing to provide assets in the pool and accept trades in exchange of a fee. The supervisors of the game are the *Arbitrageurs*. As in traditional finance arbitrageurs exploit possible differences in the price of a reference market and the pool price of an asset to make profit for free. We call them supervisors as they maintain the two prices of the assets (pool and reference price) in a relative balance. For example, if the pool price is less than the reference market price, arbitrageurs will buy the asset on the pool and sell it on an order book-based exchange for a profit.

The term *constant* refers to the fact that the given mathematical relationship of the pool's remains constant if its constant is also unchanged. A reasonable property for these curves is to be convex and monotonically decreasing because this will ensure that the price of a token is monotonically decreasing as a function of its reserves in the pool, as should be expected of a typical supply curve.

2.1.1. Price of assets & Pool value. A plot of an asset versus its price, is called *bonding curve* and shows the variation of the asset's price in the liquidity pool. Given a CFMM curve and a pool state of reserves, we can derive the *price* of a token X (in terms of a token Y) at this state as follows:

$$P_X = -\frac{dy}{dx}$$

We can see that the price of an asset is actually the slope at the given state of the pool on the curve.
In this framework, we can also define the *value of a given liquidity pool* (measured in terms of Y) as follows:

$$V_p(x, y) = P_X \cdot x + y$$

2.1.2. Constant Product Market Makers. A constant product market maker, first implemented by Uniswap, in the 2D case (2-asset pool) satisfies the equation:

$$x \times y = k$$

Where $x > 0$ and $y > 0$ are reserves of assets X and Y respectively and k is a constant. Trading an amount of Δ_x for an amount Δ_y the reserves must change in a way that their product will remain equal to the constant k. This means that every trade in CPMMs must satisfy the following equation:

$$(x + \gamma\Delta_x) \times (y - \Delta_y) = k, \quad \forall \Delta_x, \Delta_y > 0$$

where γ is the transaction fee. In practice, because Uniswap charges a 0.3% trading fee that is added to reserves, each trade actually increases k [12].
After each transaction the reserves are updated in the following way:

$$(x_{new}, y_{new}, k) = (x+\Delta_x, y-\Delta_y, (x+\Delta_x)\times(y-\Delta_y))$$

2.1.3. Constant Sum Market Makers. The simplest CFMM is the constant sum market maker (CSMM). For reserves x, y for X and Y assets a CSMM holds the sum of the reserves constant

$$x + y = k$$

Contrary to CPMMs where the slope of the curve is different at every point, and thus the price varies every

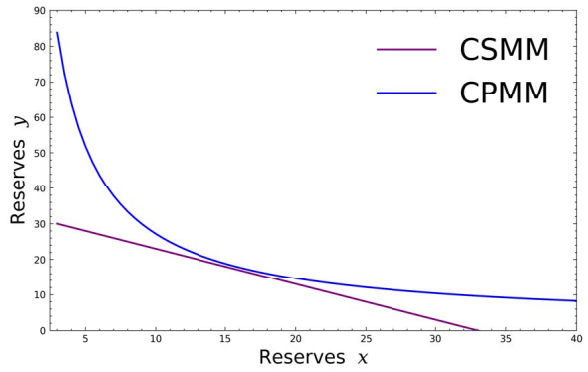

Figure 1. The constant sum (CSMM) and constant product (CPMM) functions for $k = 10$. CPMM forms a hyperbola, whereas CSMM a straight line when plotting the reserves of two assets.

time the reserves change, CSMMs $x + y = k$ keep the pricing function equal to a constant number:

$$P_X = -\frac{dy}{dx} = 1$$

Therefore, CSMMs are usually an attractive CFMM to the AMM players due to the price stability they offer, which is useful when trading tokens with stable relative prices (e.g. stablecoins) as this minimizes slippage. However, there are multiple concerns around CSMMs when an external market price is variable. In this way, arbitrageurs will be able to constantly take advantage of the price gap between the two markets and because the constant price in the CSMM cannot, by definition, adapt to situations like this, they may drain the CSMM of one of its tokens.
Figure 1 illustrates the curves of both the constant sum and constant product functions.

2.1.4. Constant Mean Market Makers. A constant mean market maker is a generalization of a constant product market maker, allowing for more than two assets of equal weights. First introduced in Balancer [10], constant mean markets satisfy the following nD equation in the absence of fees:

$$\prod_{i=1}^{n} x_i^{w_i} = k$$

where x_i are the reserves of asset $x_1, x_2, ..., x_n$, $w \in \mathbb{R}^n$ are the weights associated with each asset and $k \in \mathbb{R}_+$ the constant product. The constant mean markets ensure that the weighted geometric mean of the reserves, x_i for $i = 1, ..., n$, stays constant. In this case the weights should all satisfy $w \geq 0$ and $\sum_{i=1}^{n} w_i = 1$. Similarly to the constant product markets when trading an amount Δ_p of asset X_p for some amount Δq of a different asset X_q, where $q \neq p$ the following equation should always be satisfied:

$$\Big(\prod_{\substack{i=1 \\ i \neq p,q}}^{n} x_i^{w_i} \Big) (x_p + \gamma_p \Delta_p)^{w_p} (x_q - \Delta_q)^{w_q} = k$$

where $(1 - \gamma_p)$ is the percentage fee associated with trading asset X_p. Note here that we can retrieve the constant product market formula as a special case when we set $n = 2$ and $w_1 = w_2 = \frac{1}{2}$ and $\gamma_1 = \gamma_2$ with $k = \sqrt{k}$.
Finally, for the bileteral trading price of an asset X_i in terms of an asset X_j the following holds:

$$P_{X\,i} = \frac{w_i}{w_j} \frac{x_j}{x_i}$$

2.1.5. Hybrid CFMMs. StableSwap/Curve [9] proposed a CFMM curve that is a blend of Constant Sum and Constant Product to provide continuous liquidity, price stability and a built-in pool balancing mechanism. This function acts as a constant sum when the portfolio is balanced and shifts towards a constant product as the portfolio becomes more imbalanced. In effect, the function looks like a "zoomed-in hyperbola"

$$An^n \sum_{i=1}^{n} x_i + D = ADn^n + \frac{D^{n+1}}{n^n \prod_{i=1}^{n} x_i}$$

Finally, the authors in [14] proposed the Constant Ellipse Curve AMM with the general form of:

$$(x - \alpha)^2 + (y - \beta)^2 + bxy = k$$

in which α and β are constants. One can choose between the concave and the convex curve in the first quadrant.
These hybrid formulas are the ones that motivated our work in the sense that exhibit a larger class of automated market maker mechanisms that can include and combine multiple other CFMMs as special cases.

2.1.6. Slippage & Impermanent loss. *Slippage* as a function expresses the difference between the expected and the actual trade execution price. In the case of AMMs slippage is the loss incurred by dissimilarity of the spot price of an asset in the pool (i.e. the price that the trader sees before the trading) and the effective price obtained after the completion of the trade. On the other hand, *Impermanent loss* occurs when liquidity providers pull out assets from the pool during a large price swing. In this case they will suffer a loss of total asset value, compared to simply holding the assets. As we saw before, when a trade happens the price of an asset may change as we move from p_0 in the (x_0, y_0) state to p_n to the new state (x_n, y_n). Thus, the overall

value of the pool may change and potentially decrease. This exact decrease its what is called impermanent loss and is given by:

$$\mathcal{IL} = \frac{V_{p_n}(x_n, y_n) - V_{p_n}(x_0, y_0)}{V_{p_n}(x_0, y_0)}$$

3. λ - Constant Function Market Makers

In this section we introduce a new generalized family of constant function markets, the λ - constant function markets (λCFMMs) and we also prove that one can derive some of the traditional constant function market makers from it as special cases. The primary goal for the rest of the paper is to motivate this family of functions as it forms a novel method for blending the benefits of different CFMMs. The proposed function is a concave function that was initially inspired by the *constant power sum formula* which was first proposed in [13]. Suppose we have a pool of n assets, then the nD λCFMM defined as follows:

$$f_\lambda(\mathbf{x}) = \sum_{i=1}^{n} w_i^\lambda x_i^{1-\lambda}$$

where x_i are the reserves of asset X_i for $i = 1, 2..., n$, $w \in \mathbb{R}$ and $w_i \in \{[0, 1] : \sum_{i=1}^{n} = 1\}$ are the weights associated with each asset, $\lambda \in [0, 1)$ is the parameter of the function and $k \in \mathbb{R}^+$ is the constant.

3.1. Pricing on λCFMMs

As we saw in Section 2.1.1 the ratio $-\frac{dy}{dx}$ represents the price at which the trader sells the asset. Thus we can prove the following Theorem for λCFMMs:

Theorem 1. *The bileteral pricing function of an asset X (in terms of an asset Y) in λCFMMs is given by the following equation:*

$$P_X(\lambda) = \left(\frac{w_x y}{w_y x}\right)^\lambda$$

Proof. We want to prove that we can retrieve the λCFMMs formula from the following equation:

$$-\frac{dy}{dx} = \left(\frac{w_x y}{w_y x}\right)^\lambda$$

Indeed:

$$\Rightarrow -\frac{dy}{dx} = \frac{(w_x y)^\lambda}{(w_y x)^\lambda} \Rightarrow -w_y^\lambda y^{-\lambda} dy = w_x^\lambda x^{-\lambda} dx$$
$$\Rightarrow -w_y^\lambda \int y^{-\lambda} dy = w_x^\lambda \int x^{-\lambda} dx$$
$$\Rightarrow -w_y^\lambda \frac{y^{1-\lambda}}{1-\lambda} + c_0 = w_x^\lambda \frac{x^{1-\lambda}}{1-\lambda} + c_1$$

If we let $(c_0 - c_1)(1 - \lambda) = k$ then we get:

$$w_x^\lambda x^{1-\lambda} + w_y^\lambda y^{1-\lambda} = k$$

which indeed is the λCFMMs formula. □

3.2. Trades on λCFMMs

It is always important to define the equation that each trade needs to satisfy in order for it to be valid. For the nD case, trading A_p amount of an asset X_p for some amount A_q of a distinct asset X_q, where $q \neq p$, should always satisfy the following equation:

$$w_p^\lambda (x_p + \gamma_p A_p)^{(1-\lambda)} + w_q^\lambda (x_q - A_q)^{(1-\lambda)} + \\ + \sum_{\substack{i=1 \\ i \neq p,q}}^{n} w_i^\lambda x_i^{(1-\lambda)} = k$$

where $(1 - \gamma_p)$ is the percentage fee associated with trading asset X_p. The corresponding reserves, x_p and x_q are updated as as follows:

$$x_p \to x_p + A_p, \quad x_q \to x_q - A_q$$

By taking the second derivative of the $f_\lambda(\cdot)$ function we can see that is a concave function, and satisfies the basic properties we mentioned at the start of the Section, regarding the trades and the liquidity stability of the pool, to be considered a valid AMM function.

3.3. Traditional CFMMs as special cases

The λCFMMs can be seen as a generalized family of functions which contains certain commonly used AMM functions as special cases. Specifically, we can retrieve the constant sum, mean and product given certain values of λ.

Theorem 2. *The function f_λ can be reduced to the constant sum and constant mean formulas, as follows:*

$$f_\lambda(x, y) = \begin{cases} \sum_{i=1}^{n} x_i, & \lambda = 0 \\ \prod_{i=1}^{n} x_i^{w_i}, & \lambda \to 1 \end{cases}$$

Furthermore, for the pricing formula of an asset X:

$$P_X(\lambda) = \begin{cases} 1, & \lambda = 0 \\ \frac{w_x y}{w_y x}, & \lambda \to 1 \end{cases}$$

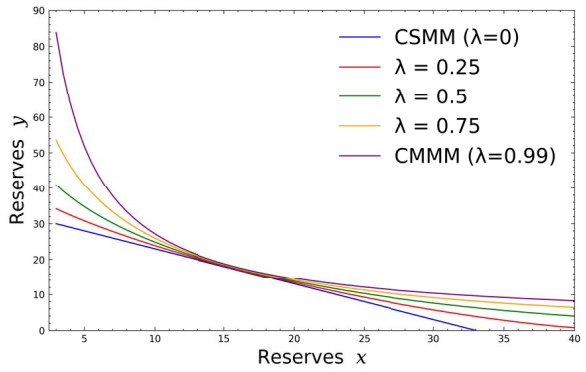

Figure 2. Liquidity curves for asset swaps for multiple values of λ for a fixed constant k. For $\lambda = 0.99$ the curve represents the constant mean (weighted product) function and for $\lambda = 0$ the constant sum function. The intermediate curves increase their curvature as $\lambda \to 1$.

In the Appendix we provide the proof for the $\lambda \to 1$ case as the rest should be straightforward. Finally, note that for $n = 2$ we can also retrieve the 2D constant sum formula and if we additionally let $w_x = w_y = \frac{1}{2}$, and $k = \sqrt{k}$ we retrieve the 2D constant product formula. In Figure 2 we illustrate the curves of λCFMM for different values of λ.

3.4. λ as the Mixture coefficient

It is clear that the family of λCFMMs is a parametrized blending of constant sum and constant mean functions. Thus, naturally we can think of λ as the mixture coefficient between these two. Consequently, λ = 0 or $\lambda \to 1$ indicates no mixture, whereas $\lambda = 0.5$ indicates the maximum mixture of these functions (i.e. you have 50% of each). Thus, we define *low mixture areas (LM)* as the ones that are close to the edges of the $[0, 1)$ range and *high mixture areas (HM)* as the ones that are concentrated around 0.5. We also, further divide the LM areas into *left low mixture area (LLM)*, as the one that is close to zero (i.e. CSMM) and *right low mixture area (RLM)*, as the one that is close to 1 (i.e. CMMM).

4. Experiments & Results

In this section we motivate the use of λCFMMs as we believe that such a generalised and potentially dynamic family of functions could be able to create mutual profits for both the traders and the liquidity providers by adjusting its formula based on the optimal parameter that minimizes *Slippage ($\mathcal{S}$)* and *Impermanent Loss ($\mathcal{IL}$)*. We conduct a number of experiments to compare the performance of λCFMMs for different mixture coefficient values as these provide different behaviors and blendings in between the two extremes of CSMM and CMMM formulas. In the first part we present the behaviour of λCFMMs by performing a grid-search for different values of λ in the $[0, 1)$ range over a trading period and in the second part we motivate the selection of optimal mixture coefficients given the same objectives. In this direction, we present a straight comparison with Uniswap v2, which can be retrieved from the λCFMMs formula for $\lambda \to 1$ and $w_i = \frac{1}{2}$.

4.1. Dataset

Our experiments examine four different pools/pairs of tokens, two of which where stablecoin pairs (USDC-USDT, DAI-USDT) and the other two with differences in price (USDC-WETH, USDT-WETH). We split our analysis both on the nature of the pool (stable/ non-stable) but also on the presence or not of transaction fees[1]. Our dataset was based on the Uniswap v2 trading trends. Specifically, for every pair of tokens we pulled Uniswap v2 data, by continuously fetching swaps for the specific pair from the chain every 1 minute. For each transaction, we got the balance of each token in the pool and the trading amounts on each block. Figure 4 provides a snapshot of the data we had and highlights the columns we used.

X	Δ_x	Y	Δ_x	R_x	R_y
USDC	-8777	WETH	+3.4059	$6.943e^7$	57300
USDC	+1854	WETH	-0.7187	$6.944e^7$	57299
USDC	+4514	WETH	-1.7500	$6.944e^7$	57298

Figure 3. We used the initial pool reserves (R_x, R_y) from the first recorded block as the initial state of our pools and use the trading demands of one of two coins (Δ_x) and update the reserves based on the λCFMMs formula for different values of λ.

4.2. Simulations

For each trade based on the input/output volume we compute the new state of the pool's reserves and also the values of slippage and impermanent loss based on the closed forms we introduced in Section 2.1.6. Finally, we track their variation on a sequence of trading blocks of the chain. The different values of λ lead to different traces on the plot indicating the different effect of the mixture coefficient. In Figures 5, 6 and 7 we

1. In the experiments that fees were included, we define the transaction fee to be $\phi = 0.25\%$.

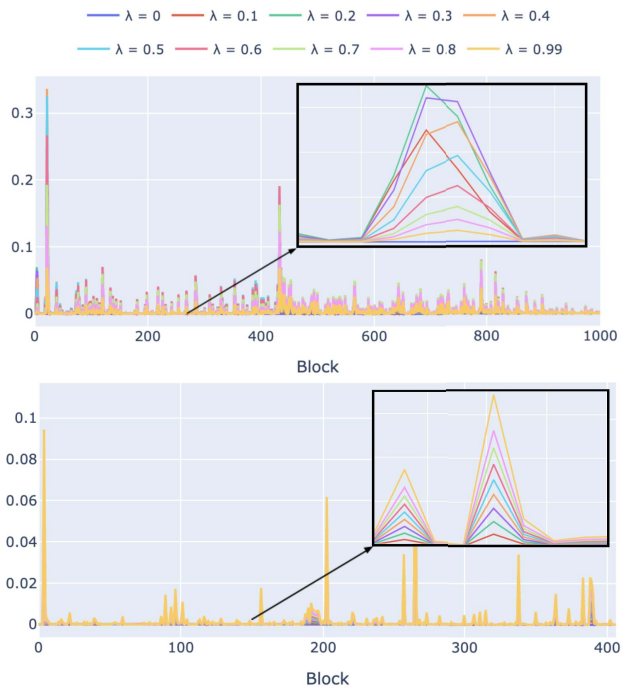

Figure 4. *Slippage variation (y-axis) for different values of λ between different blocks (x-axis) throughout a trading period.* **Top:** The non-stablecoin case of USDC-WETH pool. **Bottom:** The stablecoin case of USDC-USDT. The black frames are zoomed versions for a subset of blocks from a certain period.

can see the simulation for both metrics alongside some zoomed versions that will help the reader understand the underlying behaviour of the different functions given the nature of the pool.

4.3. Results & Remarks

We divide our analysis into three different mixture areas as follows:

$$HM = [0.35, 0.65], \quad LLM = [0, 0.35], \quad RLM = [0.65, 1)$$

4.3.1. The effect of λ on slippage variation. For the Slippage plots we can make the following remarks given the stability of the pool:
For stable pools, we observe that when the value of λ increases then the slippage follows. This is perfectly illustrated on the zoomed frame of Figure 5 (bottom). *Thus, LLM areas have the lowest slippage[2] and RLM areas the higher.*
The maximum mixture coefficient returns a slippage

2. We should keep in mind that $\lambda = 0$ represents the CSMM which always has zero slippage since the price of the assets is always stable.

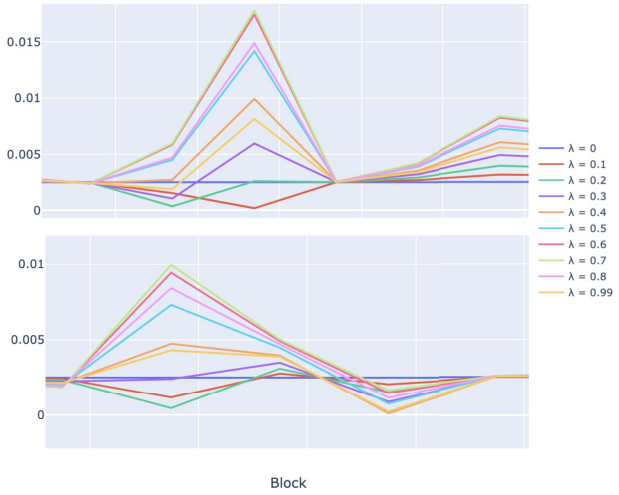

Figure 5. A zoomed version for a subset of blocks of *Slippage variation (y-axis) for different values of λ between different blocks (x-axis) throughout a trading period in the presence of transaction fees ($\phi = 0.25\%$).* **Top:** The non-stablecoin case of USDT-WETH pool. **Bottom:** The stablecoin case of DAI-USDT. The black frame is a zoomed version for a subset of blocks from the same period.

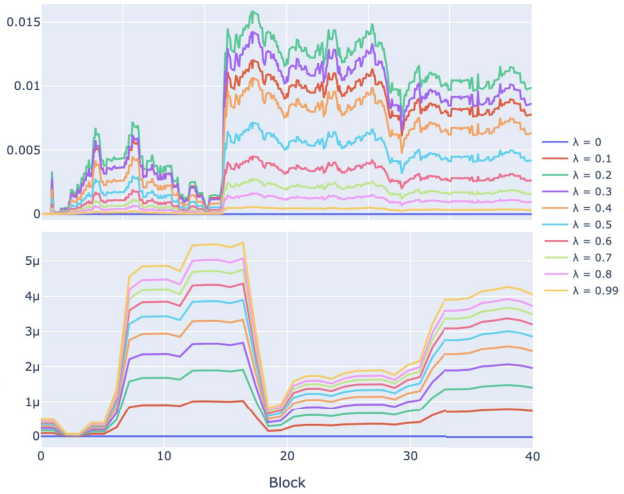

Figure 6. *Cumulative Impermanent Loss variation (y-axis) for different values of λ between a subset of blocks (x-axis) throughout a trading period.* **Top:** The non-stablecoin case of USDC-WETH pool. **Bottom:** The stablecoin case of USDC-USDT.

value in the middle. However, the addition of transaction fees breaks this ordering. Specifically, as Figure 6 (bottom) indicates the CSMM function ($\lambda = 0$), which was previously the best performer is now a straight line on the 0.0025 value ($\phi = 0.25\%$) and in many cases other values result in less slippage. This shows that:
In the realistic setting where fees are added there are mixture coefficients that blend CSMM and CMMM in a way that provides less slippage than the original

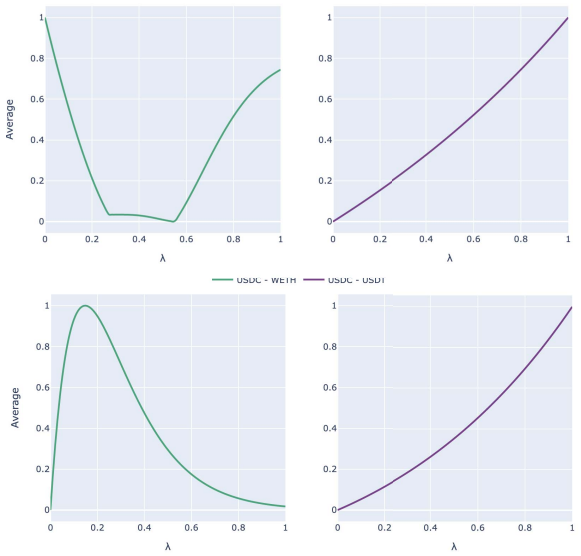

Figure 7. *Average Slippage (top) and Impermanent Loss (bottom) for 1000 values of λ in the $[0, 1)$ range*. The green line represents the USDC-WETH pool and the purple the USDC-USDT one. Transaction fee: $\phi = 0.25\%$. The data is normalized in the $[0, 1]$ range.

formulas in certain trading scenarios.

For the non-stable pools, again $\lambda = 0$ gives the minimum slippage for the same reasons, but as mentioned in Section 2 this can be dangerous since the pool is not constituted of stablecoins and arbitrageurs can easily drain it. Both in the presence of fees (Figure 5 - top) and not (Figure 6 - top) contrary to the stable case there is no ordering of the functions, meaning that different values of λ provide the optimal slippage at different blocks. Furthermore, we notice that: *In the majority of the simulations, mixture coefficients that lie on the LM areas seem to provide better results compared to the ones in the HM one.* The maximum mixture again stays in the middle and presents the less slippage fluctuation compared to the others.

4.3.2. The effect of λ on impermanent loss variation. For the Impermanent Loss things are simpler as the results are not affected by the presence of fees.

For stable pools, in Figure 7 (bottom), we observe the same ordering as the one in slippage: $\lambda \uparrow$ then $\mathcal{IL} \uparrow$.

For non-stable pools,in Figure 7 (top), we notice that in the RLM areas $\mathcal{IL}$ seems to have its best values whereas in the LLM areas the worst ones.

4.3.3. Optimal mixture coefficients. Now, in order to motivate the selection of better mixture coefficients we present the results regarding the average of the metrics for the different values of λ. Through an extensive grid search of 1000 λ-values in the $[0, 1)$ range in Figure 8 we plot the average slippage (top) and impermanent loss (bottom) of the trades for each lambda on the grid. As expected, **for stable pools** the monotonically increasing behaviour of both for $\mathcal{S}$ and $\mathcal{IL}$, illustrated in Figure 8, proves that $\lambda = 0$ (CSMM) is indeed the best choice for these pools even though we found certain trades that this would be sub-optimal in the presence of fees (Figure 6).

However, **for non-stable pools** although previously the plots indicated that low mixture could result in better slippage values, the numerical results show that:

Values of λ that lie on the HM area return the minimum average slippage.

One can clearly see this in Figure 8 (top). This occurs due to the fact that even though λ in LM areas returns seemingly good results for the majority of trades, in the ones that they are sub-optimal their respective slippage is extremely big compared to the ones in the HM area. We can now understand the benefit of the less-fluctuating behaviour of λ values in the HM area. Indeed, if we compute the standard deviation of the three areas we can confirm this claim:

$$\sigma_{\text{HM}} = 0.00005, \quad \sigma_{\text{LLM}} = 0.003, \quad \sigma_{\text{RLM}} = 0.001$$

For Impermanent Loss, by looking at Figure 8 (bottom), the remarks of visual and numerical results match completely as we notice that maximum $\mathcal{IL}$ can be found in the LLM area and gets better as λ approaches the RLM area.

Finally, in the table below we analytically compute the optimal value of λ (λ^*) and the respective slippage (s^*) and demonstrate the superiority of λCFMMs compared to Uniswap in all the different cases. The optimal values are the ones that return the minimum average slippage and are computed as follows:

$$\lambda^* = \min_{\lambda} \overline{\mathcal{S}(\lambda)}$$

where

$$\overline{\mathcal{S}(\lambda)} = \frac{1}{n-1} \sum_{i=2}^{n} \mathcal{S}_n(\lambda) \quad \text{and}$$

$$\mathcal{S}_n(\lambda) = \mathcal{S}_n(x_{n-1}, y_{n-1}, \Delta_x^n)$$

for Δ_x^n indicating the n-th trade on a given pool.

Pool	λCFMM		Uniswap	
	$\lambda*$	$s*$	$\lambda*$	$s*$
USDC - WETH	0.545	**0.3245**	0.999	0.3325
USDT - WETH	0.407	**0.3346**	0.999	0.3350
USDC - USDT	0.0	**0.3339**	0.999	0.3375
DAI - USDT	0.0	**0.3322**	0.999	0.3403

Overall, our results suggest that regarding $\mathcal{S}$, optimal functions for non-stablecoins seems to be the ones with high mixture coefficient (i.e. the ones that strongly blend CSMM and CMMM), whereas for stablecoins CSMM always provides the optimal formula.

5. Conclusions

This work introduces a generalized AMM function called λCFMM from which popular AMM functions can be retrieved as special cases. The numerical results alongside the simulations' plots, over a a trading period, indicate the benefits of the proposed formula in targeting more efficient mixed functions in terms of slippage and impermanent loss, compared to the standard markets. Overall, λCFMMs' structure offers a novel method for blending the benefits of constant sum and constant mean market makers and opens the path for future work, which will have to do with the designing of dynamic AMM functions, that given certain features, could be able to self-adjust their parameters and simultaneously benefit the traders and the liquidity providers.

References

[1] M. Wyart, J.-P. Bouchaud, J. Kockelkoren, M. Potters, and M. Vettorazzo, "Relation between bid–ask spread, impact and volatility in order-driven markets.", Quantitative Finance, vol. 8, no. 1, pp. 41–57, 2008.

[2] A. Juliano, "dydx: A standard for decentralized derivatives," 2017.

[3] W. Warren and A. Bandeali, "0x: An open protocol for decentralized exchange on the Ethereum blockchain," 2017.

[4] P. Daian, S. Goldfeder, T. Kell, Y. Li, X. Zhao, I. Bentov, L. Breidenbach, and A. Juels, "Flash Boys 2.0: Frontrunning, Transaction Reordering, and Consensus Instability in Decentralized Exchanges," arXiv:1904.05234 [cs], Apr. 2019.

[5] E. Hertzog, G. Benartzi, and G. Benartzi, "Bancor protocol:continuous liquidity for cryptographic tokens through their smart contracts," tech. rep., 2018.

[6] Robert Winkler. "Scoring rules and the evaluation of probability assessors." Journal of the American Statistical Association, 64(327):1073–1078, 1969.

[7] R. Hanson, "Logarithmic markets scoring rules for modular combinatorial information aggregation," The Journal of Prediction Markets, vol. 1, no. 1, pp. 3–15, 2007.

[8] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 core," tech. rep., 2020.

[9] Michael Egorov. StableSwap - efficient mechanism for Stablecoin liquidity (2019)

[10] F. Martinelli and N. Mushegian, "Balancer: A non-custodial portfolio manager, liquidity provider, and price sensor," 2019

[11] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. Uniswap v3 core. Technical report, 2021.

[12] Guillermo Angeris, Hsien-Tang Kao, Rei Chiang, Charlie Noyes, and Tarun Chitra. An analysis of Uniswap markets. Cryptoeconomic Systems, November 2019.

[13] Allan Niemerg, Dan Robinson, and Lev Livnev, Yieldspace: An automated liquidity provider for fixed yield tokens, (2020).

[14] Y. Wang, "Automated market makers for decentralized finance (defi)," 2020.

Appendix

As we saw Theorem 2 states the following:

$$\text{For } \lambda \to 1 \Rightarrow w_x^\lambda x^{1-\lambda} + w_y^\lambda y^{1-\lambda} = x^{w_x} y^{w_y}$$

Proof. We will show that as λ approaches 1, then λCFMM approaches the constant mean formula. Here we are proving it in the case where $n = 2$ but one can easily see the derivations for $n > 2$.
First we rewrite the constant k in terms of the constants x_0 and y_0 as follows:

$$f_\lambda(x, y) = f_\lambda(x_0, y_0) = k$$
$$w_x^\lambda x^{1-\lambda} + w_y^\lambda y^{1-\lambda} = w_x^\lambda x_0^{1-\lambda} + w_y^\lambda y_0^{1-\lambda}$$

Let $w_x^\lambda = \alpha$ and $w_y^\lambda = \beta$

$$\Rightarrow y = \left(\frac{\alpha x_0^{1-\lambda} + \beta y_0^{1-\lambda} - \alpha x^{1-\lambda}}{\beta} \right)^{\frac{1}{1-\lambda}}$$

$$\Rightarrow \ln y = \frac{1}{1-\lambda} \ln \left(\frac{\alpha x_0^{1-\lambda} + \beta y_0^{1-\lambda} - \alpha x^{1-\lambda}}{\beta} \right)$$

When $\lambda \to 1$, the right side of the equation is an undetermined form of $\frac{0}{0}$ and also $\alpha = w_x^\lambda \to \alpha' = w_x$ and $\beta = w_y^\lambda \to \beta' = w_y$. Thus, by applying L'Hopital's rule gives us the following limit:

$$\lim_{\lambda \to 1} \frac{\frac{\partial}{\partial \lambda} \ln(\alpha' x_0^{1-\lambda} + \beta' y_0^{1-\lambda} - \alpha' x^{1-\lambda}) - \frac{\partial}{\partial \lambda} \ln \beta'}{\frac{\partial}{\partial \lambda}(1-\lambda)}$$

$$= -\lim_{\lambda \to 1} \frac{-\alpha' \ln x_0 x_0^{1-\lambda} - \beta' \ln y_0 y_0^{1-\lambda} + \alpha' \ln x x^{1-\lambda}}{\alpha' x_0^{1-\lambda} + \beta' y_0^{1-\lambda} - \alpha' x^{1-\lambda}}$$

$$= -\frac{-\alpha' \ln x_0 - \beta' \ln y_0 + \alpha' \ln x}{\beta'}$$

$$\Rightarrow \alpha' \ln x + \beta' \ln y = \alpha' \ln x_0 + \beta' \ln y_0$$

$$\Rightarrow \ln x^{\alpha'} + \ln y^{\beta'} = \ln x_0^{\alpha'} + \ln y_0^{\beta'}$$

$$\Rightarrow e^{\ln x^{\alpha'} + \ln y^{\beta'}} = e^{\ln x_0^{\alpha'} + \ln y_0^{\beta'}}$$

Using the sum property of the logarithm and by replacing the values of α' and β' we finally get:

$$x^{w_x} y^{w_y} = f_\lambda(x_0, y_0) = k$$

which is the constant weighted product formula. □