Derivio: On-chain Derivatives Market Making Using Adaptive Curves

Derivio Labs

February 2023

1 What is Derivio?

Derivio is a Liquidity-as-a-Service (LaaS) derivatives protocol suite, offering synthetic derivatives that granularize risk-adjusted rewards for traders with smart leverage, while providing operators with sufficiently deep liquidity pools for effective token hedging.

In Phase 1, Derivio introduces perpetual derivatives and digital options with aggregated liquidity over fungible tokens, NFT, forex, and commodity markets. Liquidity-as-a-Service tokenization is also used for interest rate, zero coupon bond, and index derivatives markets — novel composable DeFi primitives.

Derivio aims to create a forward-thinking DeFi derivatives ecosystem, building for where DeFi and its users are headed – not where the space currently is. By building fair & highly liquid derivatives markets with one-click abstraction for market-neutral derivatives market making, Derivio aims to achieve a high level of UX simplicity and a broad range of products wide enough for retail and institutional participants of varying needs and expertise.

2 Oracle-based Derivatives Model (ODM)

2.1 A Brief Explanation of ODM

The position opening price and closing price come from the oracle feed(s). The liquidity pool serves as the counter-party to active traders, effectively acting as the sole market maker. The liquidity pool receives revenue from trading fees and distributes to liquidity providers. Here is a short example. For simplicity, we exclude any fees incurred:

Trader Alice comes in and deposits 1,500 USDC. Assuming the ETH price is $1,500, Alice opens a short position with 10x leverage. The pool will lock up

15,000 USDC, since this is the maximum profit from a 10x position. If at some point ETH drops to 1,200 USDC and Alice chooses to exit the position, 3,000 USDC from the 15,000 locked USDC fund will be credited to Alice's account, with the other 12,000 USDC unlocked and released back to the pool.

Here is another example: trader Bob comes in and deposits 1,500 USDC to open a long ETH position with 10x leverage. This time, the pool will lock 10 ETH - again, this is the maximum potential profit for the trader. If ETH price surges to $2,000, Bob would happily take the 5,000 USDC (or, 2.5 ETH) profit, with the remaining 7.5 ETH unlocked and released back to the pool.

2.2 Why ODM?

There are several key features that makes the ODM stand out:

- ODMs are able to provide **zero-slippage trading** regardless of the order size and liquidity depth because they receive price feeds from oracles.
- The liquidation process is simple and **does not involve selling any coins in the market**. This prevents settlement problems when the price experiences sudden high volatility.
- **Instant Liquidity** available. In traditional order-book-based perpetual exchanges, long and short open positions should match each other for trade to be executed. The existence of a liquidity pool with oracle price feed enables unbalanced long-short open interest and gives traders enough liquidity at any time they want.

2.3 Challenges that ODM faces

Existing ODM models are far from perfect – they face several significant challenges, namely:

First, slippage manipulation: As with most things, there is no free-lunch. The same goes for zero slippage! Every single trade that traders make should technically have some impact on price or extra costs paid to the market/counterparties (albeit minimal most of the time). Otherwise, in existing ODM designs, a trader can take advantage of zero slippage trading and exploit the liquidity pool (e.g AVAX exploitation on GMX).

Second, imagine a one-sided market condition. Since the liquidity pool is acting as the passive market maker (counter-party), if there were no mechanisms to balance out the extreme long-short ratio, active traders can exploit the liquidity pool (e.g. gTrade vault collateralization event during LUNA's collapse).

3 Market Neutral Liquidity for ODM

Both of these problems points to an urgent need - DeFi needs an additional balancing mechanism to avoid or mitigate potential risks in these situations. In an efficient derivatives market, the more traders take position on one side of the market, the more expensive it is to take position on that side of the market, and the more profitable it is to take the opposite side. This ensures that the long-short ratio can be balanced between active traders under extreme market conditions by creating arbitrage opportunities. To further illustrate how this protects the liquidity pool, we first introduce the concept of market neutrality.

In finance, **market neutrality** is a risk-minimizing strategy that entails portfolio managers structuring their portfolio so that they would gain regardless of which direction the market is moving.

Market-Neutral Liquidity refers to the ability for liquidity pools to self-rebalance and converge to the market-neutral state under one-sided or volatile market conditions, thus providing robust protection for liquidity providers while generating organic yield from market-making.

3.1 Live funding rate for perpetual futures

We propose a funding rate paid between active traders, with a novel dynamic funding curve design based on long-short ratio, which is the most effective indicator for bid/ask market condition:

$$\text{Long \%} = \frac{\text{Long OI}}{\text{Long OI} + \text{Short OI}}$$

$$\text{Short \%} = \frac{\text{Short OI}}{\text{Long OI} + \text{Short OI}}$$

The funding rate is a risk management measure, aiming to ensure that the platform remains stable, sustainable, and robust. Thus, the funding curve should not have too significant an impact on trading experiences – it should only become prominent when significant open interest imbalance happens.

We hereby introduce an upper threshold t_{up} and a lower threshold t_{down}. Only when the long percentage is outside these thresholds do we take it into account for the (raw) position adjustment constant:

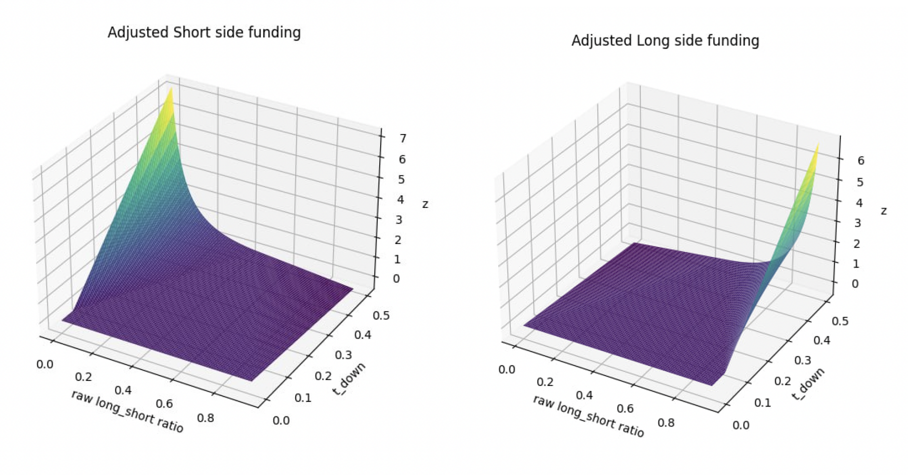
$$\text{Raw adjustment} = \begin{cases} \text{Long \% - } t_{up}, & \text{if Long \%} > t_{up} \\ 0, & \text{if } t_{down} < \text{ Long position \%} < t_{up} \\ \text{Long \% - } t_{down}, & \text{if Long \%} < t_{down} \end{cases}$$

In order to maintain symmetry and minimize pool exposure to net long or short position, t_{up} will usually be set as $1 - t_{down}$. Thus:

$$\text{Raw adjustment} = \max(\text{Long }\%,\ t_{up}) + \min(\text{Long }\%,\ t_{down}) - 1$$

t_{down} will therefore be a constant between 0 and 0.5; when $t_{down} = 0$, the positional adjustment is always 0, and hence no funding will be collected; on the other hand, when t_{down} approaches 0.5, the positional adjustment will be just equal to long percentage - 0.5. This parameter should thus be fine-tuned according to the risk volatility, with *volatile markets using higher positional adjustment*.

Also, since the open interest of the long and short sides might not be the same, applying the same funding rate will imply more funding collected than given out. In order to balance this, we use the following formula to distribute them fully, which is also shown in the figure:



(a) Short side

(b) Long side

Figure 1: Adjusted funding curves

$$\textbf{Position adjustment} = \text{Raw adjustment} \times \max\left(1,\ \frac{\text{open interest of opposite side}}{\text{open interest of this side}}\right)$$

With this in hand, we can finally introduce our live funding rate, which consists of 3 parts:

- **Position adjustment**. The long-short imbalance indicator we discussed above.

- **Borrow rate**, defined as $\frac{\text{open interest}}{\text{total assets available}}$. We want to keep the funding rate reasonable during times when the open interests are low in general - since under this circumstance, any opened position, even a relatively small amount, could impact the long-short percentage significantly.
- **Regularization constant**. By adjusting this, we set a cap on how much funding is paid per hour. This parameter aims to scale the whole funding amount into an appropriate range.

Live funding rate = borrow rate × position adjustment × regularization constant

3.2 Block-based funding collection

The live funding fee is calculated per block and added to the total to-be-collected funding fee. The integration is implemented by Riemann sum on-chain in a resource efficient way.

$$\begin{aligned}\textbf{Funding fee} &= \int_0^{\text{now}} \text{traders\_position\_value}(t) \times \text{funding\_rate}(t)\ dt \\ &= \int_0^{\text{now}} \text{traders\_position} \times \text{index\_price}(t) \times \text{funding\_rate}(t)\ dt\end{aligned}$$

4 Generalized Margin System

Although the previous perpetual contract model offers trading of every coin against USD, there is a growing demand for exotic pairs. For example:

1) Crypto-crypto pair. When traders have a particular opinion on a coin and want to hedge their market trend exposure, they may prefer a market with BTC or ETH as the quote currency, such as ETH/BTC, UNI/ETH, LDO/ETH, among others. If they simply long one and short another, they will face the re-balancing issue which implies additional cost and active management.

2) stablecoin pair. The concept of "USD" is not well-defined in the crypto space. There are various stablecoins like USDC, USDT, among others, each with their own risks and potential de-pegging events. This implies that BTC/USDT and BTC/USDC would have different fair values, and treating them the same could lead to adverse selection while one of them lose the peg. Besides, traders may want to trade stablecoin pairs to hedge their exposure to particular stables or express an opinion on their safety.

A Generalized Margin System (GMS) addresses these issues by allowing traders to trade on any pair in the pool rather than limiting them to coins over USD.

4.1 from ODM to GMS

Let us re-visit the example in 2.1, but this time, Alice and Bob are trading the ETH/BTC market.

Trader Alice comes in and deposits 1 BTC. Assuming the ETH/BTC price is 0.1, and Alice opens a short position with 10x leverage. 10 BTC from the pool will be reserved, since this is the maximum profit from a 10x position. If at some point ETH/BTC drops to 0.05 and Alice chooses to exit the position, 5 BTC from the 10 locked BTC fund will be credited to Alice's account, with the other 5 unlocked and released back to the pool. Effectively, the generalized margin system offers a symmetric view of longs and shorts.

4.2 Funding payment under GMS

The funding rate between traders will continue to apply to every position, but with a generalized margin system, determining the market against which traders are trading for an arbitrary pair becomes less straightforward. Our solution is to separate each pair into a long token and a short token, and apply our previous funding model to each token separately. For example, for a long position in ETH/BTC, traders will pay or receive long funding for ETH and short funding for BTC. Conversely, for a short position in BTC/USDT, it will be short funding for BTC and long funding for USDT.

Calculating the token fundings separately requires some adjustments for coins since they are not associated with a market. In these cases, the concept of long and short open interest will be defined differently. Specifically, we have:

$$\text{Long OI of coin X} = \sum_{coin Y \in \text{pool}} \text{Long(X) OI of pair X/Y for all Y}$$

$$\text{Short OI of coin X} = \sum_{coin Y \in \text{pool}} \text{Short(X) OI of pair Y/X for all Y}$$

5 Adaptive Payout Curve for Digital Options

Existing on-chain option trading suffers from serious mispricing & liquidity fragmentation. Derivio applies the idea of market-neutral liquidity and achieves efficient pricing in on-chain option markets with instant liquidity, regardless of the spatial and temporal fragmentation of the order flow.

Again, we start with the idea of the long-short ratio. Why is this the case?

Imagine a very simple digital options market between Alice and Bob, where Alice bets down and Bob bets up. They agree on how many chips to put on the table, and the winner will get all of them. In this case, we can see the payout% is proportional to the inverse of how much you bet - if you paid 20% of the chips, your expected payout should be the other 80%, which is equivalently 4x.

Using this logic, we can write out our preliminary formula for payout:

$$\text{(raw) Long \%} = \frac{\text{Long OI}}{\text{Long OI} + \text{Short OI}}$$
$$\text{(raw) Short \%} = \frac{\text{Short OI}}{\text{Long OI} + \text{Short OI}}$$

$$\text{(raw) Long payout} = \frac{\text{Short \%}}{\text{Long \%}} \times c_{profit}$$
$$\text{(raw) Short payout} = \frac{\text{Long \%}}{\text{Short \%}} \times c_{profit}$$

where the profit constant $c_{profit} = 1 - c_{cut}$ is the total percentage that goes to the trader, and the remainder goes to the pool as commission fee.

However, since the long-short ratio may vary at any time, the payout of digital options is thus not pre-determined and varies according to the long-short ratio. We introduce an integral of long percentage over the whole period, allowing for more flexibility and dynamic pricing, whilst more accurately reflecting the market conditions and sentiments.

$$\text{Final long \%} = \int_0^{\text{settle}} \frac{\text{adjusted long \%}}{\text{settlement period length}}\, dt$$
$$\text{Final short \%} = \int_0^{\text{settle}} \frac{\text{adjusted short \%}}{\text{settlement period length}}\, dt$$

5.1 Adjusted long/shorts: covering extreme cases

You may have noticed that we have used *adjusted long %* instead of *raw long %* in our final formula above. These adjustments are to make the percentage more faithfully reflect the market condition:

Firstly, extreme long-short ratios can occur when trading activities are low, which can be harmful to traders as it can result in very high or very low payouts. To combat this effect, Derivio introduces a regularization parameter that adds the same initial amount to the long and short sides of the ratio, effectively flattening the curve.

$$\text{Long position \%} = \frac{\text{Long OI} + c_{reg}}{\text{Long OI} + \text{Short OI} + 2 \times c_{reg}}$$

$$\text{Short position \%} = \frac{\text{Short OI} + c_{reg}}{\text{Long OI} + \text{Short OI} + 2 \times c_{reg}}$$

This regularization parameter reduces the effect of outliers and creates a more stable payout structure, which ensures more consistent payouts for traders – especially for early comers, which can increase their confidence in the platform and encourage more trading activity. For example, suppose c_{reg} = \$10,000 and Long OI = Short OI = \$2,000. At this early stage, if one trader comes in with \$10,000 long, their payout goes down to less than 17% immediately. However, with the regularization parameter, their payout stays at around 54%, which is almost 3 times more.

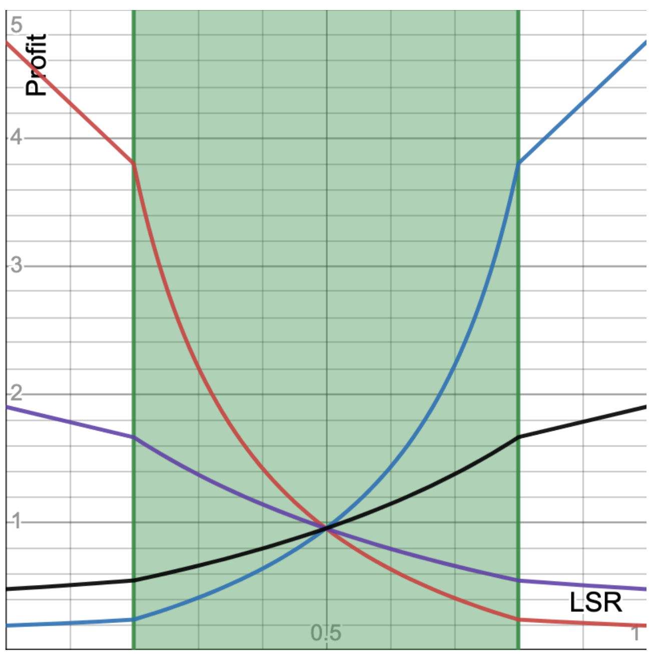


Figure 2: Regularization effect

You can play with the interactive graph here, to see how this feature is going to alter the frontier of the payout curve.

However, extreme ratios could still break in when some sort of market manipulation or one-sided markets occur. This is another case in which our protection mechanism kicks in; we will set the floor of our long/short percentage at a fixed value c_{floor}, usually 20%, to keep the whole curve stable without any sudden jumps.

Therefore, our final formulas are as follows:

$$\text{Adjusted long position }\% = \max\left(\frac{\text{Long OI} + c_{reg}}{\text{Long OI} + \text{Short OI} + 2 \times c_{reg}},\ c_{floor}\right)$$

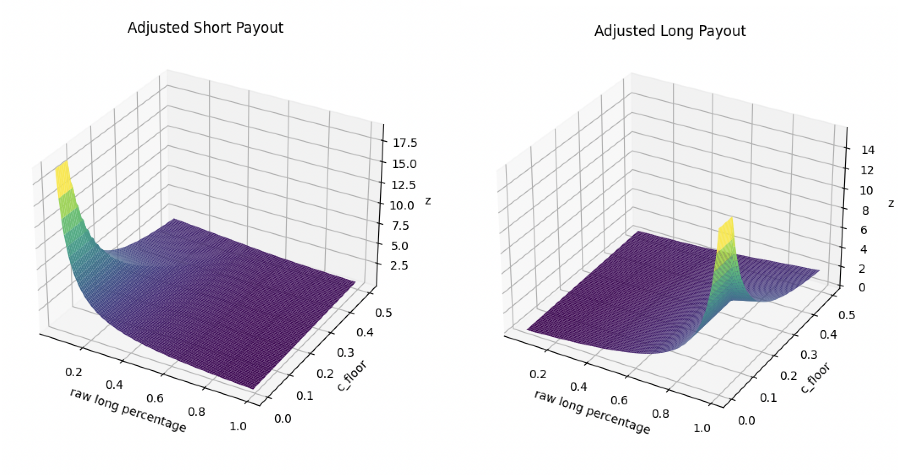
$$\text{Adjusted short position }\% = \max\left(\frac{\text{Short OI} + c_{reg}}{\text{Long OI} + \text{Short OI} + 2 \times c_{reg}},\ c_{floor}\right)$$

$$\text{Final long }\% = \int_0^{\text{settle}} \frac{\text{adjusted long }\%}{\text{settlement period length}}\, dt$$

$$\text{Final short }\% = \int_0^{\text{settle}} \frac{\text{adjusted short }\%}{\text{settlement period length}}\, dt$$

$$\text{Final Long payout} = \frac{\text{Final Short }\%}{\text{Final Long }\%} \times c_{profit}$$

$$\text{Final Short payout} = \frac{\text{Final Long }\%}{\text{Final Short }\%} \times c_{profit}$$



(a) Short side

(b) Long side

Figure 3: Adaptive payout curves

Similar to perpetual future's adaptive funding curve, the adaptive payout curve is implemented by Riemann Sum on chain.

5.2 Re-discover CFMM

As you have seen above, in contrast to perpetual derivatives where determining the asset price is sufficient, accurately pricing digital options requires much broader methodologies. While the oracle-based pricing model effectively fetches the correct strike and settlement prices, determining the payout requires lots of further careful considerations, which we have walked through.

If we view the Derivio's aforementioned model in another way, by denoting the final long payout as x, the final short payout as y, and rearranging our formula, we would have:

$$xy = \frac{\text{Final Short \%}}{\text{Final Long \%}} \times c_{profit} \times \frac{\text{Final Long \%}}{\text{Final Short \%}} \times c_{profit} = c_{profit}^2$$

Surprisingly, we discover a constant-function market maker (CFMM) hidden behind the scene! A CFMM is a market maker with the property that the amount of any asset held in its inventory is completely described by a well-defined function of the amounts of the other assets in its inventory. Its variants are commonly used as a price discovery tool in numerous on-chain exchange models. By giving traders the right to move the price on the curve through trading activities, the model could guarantee instant liquidity without distorting the price feed.

In the remaining part of this section, we will walk through our parameters once again and try to discover their actual role in Derivio's exotic CFMM variant for digital options.

The regularization constant c_{reg}, can be thought of as the initial liquidity added to both sides of the CFMM curve. When $c_{reg} = 0$, the pool has no liquidity, meaning any open orders can push the price to either 0 or infinity. By introducing c_{reg}, the pool can provide some liquidity into the $xy = c_{profit}^2$ model at a fair price of 0.5, to be adjusted by traders. Similar to the vanilla CFMM model, the higher the value of c_{reg}, the less market impact traders will have.

$$\text{Long OI} = \text{Long OI} + c_{reg}$$

$$\text{Short OI} = \text{Short OI} + c_{reg}$$

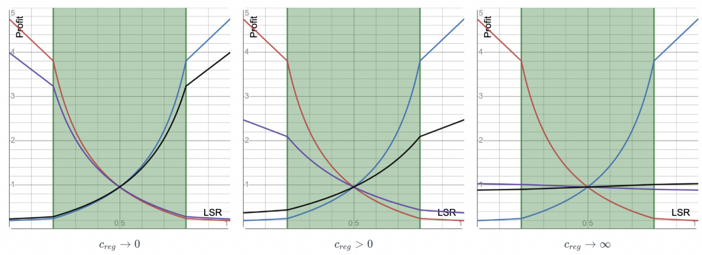


Figure 4: Regularization effect with varying parameters

The long-short ratio floor value c_{floor} is depicted in Figure 4 as the flattened section of the CFMM curve beyond the green normal range. As you can see, the curve deviates from the hyperbola and becomes linear beyond the green ranges, and the impact of trader is thus slightly reduced. By capping the payout curve within reasonable ranges, traders are able to determine the fair pricing of options without any unexpected fluctuations.

Ultimately, the integral / Riemann sum, representing the Time-Weighted Average Payout discovered over the entire trading period in the CFMM model, mitigates any adverse impact on user experience from rapid fluctuations. This ensures a fair execution environment for all traders, and could also reduce the harm of any potential oracle manipulation events.

6 Conclusion

Derivio introduces the first market-neutral liquidity model for risk-optimized, capital-efficient on-chain derivatives automated market-making. This is achieved by the introduction of adaptive funding and payout curves. Traders can trade with fair pricing over a wide range of markets, while liquidity providers (automated market makers) facilitate efficient derivatives market-making with minimal directional exposure through Derivio's novel market-neutral liquidity model.