Monday Trade Perp*

Monday Trade Team

November 19, 2025

Abstract

Monday Trade Perp is the latest iteration of the decentralized derivatives platform, designed for serving a large pool of market participants. The revolutionary fully on-chain Monday Trade Perp Engine combines the best of automated market maker and order book to provide deep liquidity, empowering all traders, order book market makers, and liquidity providers.

1 Harmonizing Simplicity and Efficiency

Fifteen years after the inception of Bitcoin, decentralized finance (DeFi) has delivered its promises of transparency, permissionlessness, and accessibility. Remarkably, this financial inclusivity has been extended to those seeking and providing services. For example, Automated Market Makers (AMMs) have democratized activities once exclusive to professionals; market making is now as straightforward as making a deposit. Based on the pioneering work of spot market such as Uniswap, Monday Trade Perp is an AMM designed for single token liquidity provision for derivatives. Now, anyone can launch a derivatives market with a single token for any asset, including major cryptocurrencies, altcoins, Non-fungible Tokens (NFTs), hash rate indices, and Real World Assets (RWAs). This innovation has diversified the range of assets and enabled non-professionals to reap yields and fees in a permissionless environment.

However, this simplicity has also sparked criticism. On the one hand, it undermines capital efficiency and, consequently, liquidity depth. On the other hand, it hampers operational efficiency. Less experienced users may struggle to grasp all the rules, risking losses, and even seasoned users may find it challenging to stay updated on every protocol and market movement.

In response to these challenges and after facilitating billions of dollars in trades, Monday Trade has increasingly realized that improving the existing financial system does not eliminate the need for professionals. Instead, an optimal system, akin to other natural systems, should be sufficiently decentralized and permissionless to welcome a wider variety of participants, transparent enough to select genuine experts, and accessible enough for everyone to benefit from the services provided by professionals, all without prohibitive intermediary costs or entry barriers.

With this conviction, we present our groundbreaking Monday Trade Perp engine, a comprehensive derivatives market-making model that combines the best of the order book and AMM models on-chain and is engineered to harmoniously blend simplicity with efficiency and tailored for both novices and experts alike.

2 Monday Trade Perp engine: Key Features

Monday Trade Perp engine has the following key features.

*This document is intended for review and comment purposes only and its content is subject to change at any time without notice. Please do not distribute or solely rely on the information here. Your feedback and suggestions for improvement are welcome and encouraged.

2.1 Single-Token Concentrated Liquidity for Derivatives

Monday Trade Perp engine supports concentrated liquidity within specific price ranges and incorporates leverage to increase capital efficiency.

Unlike prevalent spot market liquidity models such as Uniswap v3, Monday Trade Perp engine introduces a margin management and liquidation framework tailored for derivatives. Moreover, it implements the concept of two-sided liquidity while utilizing just a single token, eliminating the necessity to provide liquidity for both sides of a token pair. This streamlined approach achieves capital efficiency and maintains the permissionless nature of DeFi.

2.2 Fully On-chain Order Book

AMMs have democratized market access, offering automated "market maker" functionality even for niche assets. However, this comes at the expense of capital inefficiency as AMMs demand significant more liquidity to achieve similar price impact level as in order books.

Order books, on the other hand, don't support volatile digital assets due to complex infrastructure and risk management concerns. However, they are ideal for capital efficiency, concentrating liquidity around mid-price. With that in mind, we also implemented an order book in Monday Trade Perp.

Therefore, after rejecting off-chain and hybrid alternatives, we chose a fully on-chain approach for the order book, guaranteeing transparency, trustlessness, and anti-censorship. The model promises security and robustness by eliminating dependence on centralized administrators to process orders with the potential for "backdoors" and mitigating vulnerabilities across on-off chain systems, including order management matching and executions in alternative models.

2.3 Unified Model for Unified Liquidity

Monday Trade Perp engine introduces an innovative liquidity paradigm by seamlessly integrating concentrated liquidity and order book in a single model, offering a unified liquidity system tailored to active traders and passive liquidity providers (LPs). This cohesive approach ensures that traders can enjoy efficient atomic transactions with predictability.

Conversely, in systems with on-chain AMMs and off-chain order books or separate on-chain order books, trade requests are split between subsystems, leading to inefficiencies, non-atomic execution, and unpredictability. This dual-process execution risks non-synchronization, potentially leaving the AMM to process the transaction while the order book system falters, introducing potential confusion if the order book operates off-chain.

3 Single-Token Concentrated Liquidity for Derivatives

3.1 Synthetic AMM

Synthetic Automated Market Maker (sAMM) enables LPs to provide liquidity to a trading pair with only one type of token.

Monday Trade Perp engine represents a significant step forward in the evolution of market maker models. Before diving deeper, let's recap the plain vanilla constant product AMM. Denote by x, y, and k the quantities of token 0, token 1, and the constant product. The price of the trading pair of token 0 in token 1 is P and the total value of the tokens added is M. The relations between these variables are as follows.

$$xy = k$$
$$y/x = P$$

For the spot model, there is an additional constraint:

$$xP + y = M$$

When adding liquidity to a spot AMM, the user chooses one of x and y. The other one is implied by the smart contract with current price P.

For derivatives, the introduction of margin increases the dimension of the entire mathematical model. However, sAMM makes it almost identical to the plain vanilla constant product AMM model to flatten the learning curve and reuse existing tools. The process of adding liquidity in an sAMM to an ETH-USDC pair is:

1. LP adds the total margin of M USDC.
2. An ETH long position with value $\frac{M}{2}$ is created and added into AMM's long position inventory. (An equal amount of ETH-USDC short position is created and added into LP' s account to balance the market.)

In common AMM terms, $x = \frac{M}{2P}$, $y = \frac{M}{2}$. Compared to the vanilla AMM model for spot, the value of total token added is the same (M USDC). The difference is, in spot AMM, $\frac{M}{2P}$ ETH and $\frac{M}{2}$ USDC must be added whereas in sAMM, no changing hands between ETH and USDC happens at all.

3.2 Concentrated Liquidity Model and Margin Requirements

Unlike liquidity pools in the sAMM where there is only one AMM for all LPs in a pool, each concentrated liquidity in Monday Trade Perp engine is a homogeneous yet independent AMM. Trading volume is allocated based on the liquidity of each concentrated liquidity covering the price range. The discussions below focus on a single concentrated liquidity.

To boost capital efficiency, Monday Trade Perp engine employs concentrated liquidity. Like concentrated liquidity for the spot market, each liquidity in Monday Trade Perp only supports trading in a specific price range. We introduce helper variables x_{virtual} and y_{virtual} for the implied x and y amounts in a full range constant product AMM. Let the current price of token 0 in token 1 be P_c and the lower and upper bounds for the price range be P_a and P_b. Denote by r_i the initial margin requirement ratio. To reduce complexity for LPs and prevent one-sided liquidity from unintended liquidation, We require that

$$P_a = P_c/\alpha,\ P_b = \beta \cdot P_c,\ \alpha, \beta \geq 1 + r_i.$$

LPs are only required to specify the width of the price range instead of the lower and upper price of the range.

The process of adding liquidity to Monday Trade Perp is similar as that to sAMM, where a long position is created (x_{real}) for the liquidity and an offset short position is created for the LP.

To determine the margin required for a concentrated liquidity, we require that the initial margin be sufficient for the resulted net position at price boundaries P_a and P_b. Based on the constant product formula and the fact that x_{real} is the trade size that moves price from P_c to P_b, we have

$$x_{\text{virtual}} = x_{\text{real}} \cdot \frac{\sqrt{\beta}}{\sqrt{\beta} - 1}, \tag{1}$$

$$y_{\text{virtual}} = x_{\text{virtual}} \cdot P_c, \tag{2}$$

$$\mathsf{NetPosition}\,(P) = x_{\text{virtual}} \cdot \left(\sqrt{\frac{P_c}{P}} - 1 \right), \tag{3}$$

$$\mathsf{PnL}\,(P) = -x_{\text{virtual}} \cdot \left(\sqrt{P_c} - \sqrt{P} \right)^2, \tag{4}$$

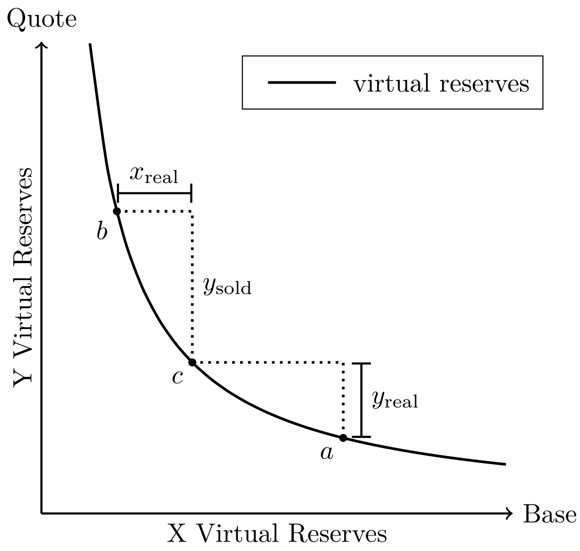

Figure 1: Virtual price curve in Monday Trade Perp

where $P \in [P_a, P_b]$ is the AMM price. At P_b, we require

$$M + \mathsf{PnL}\left(P_b\right) = \left|\mathsf{NetPosition}\left(P_b\right)\right| \cdot P_b \cdot r_i$$

Using Equation (1), (3) and (4), we have:

$$M = x_{\text{virtual}} \cdot \left(\sqrt{P_c} - \sqrt{P_b}\right)^2 + x_{\text{virtual}} \cdot \left(1 - \sqrt{P_c/P_b}\right) \cdot P_b \cdot r_i = x_{\text{real}} \cdot P_c \cdot \left[\beta \cdot (1 + r_i) - \sqrt{\beta}\right] \tag{5}$$

Similarly, at P_a

$$M' = x_{\text{real}} \cdot P_c \cdot \frac{\sqrt{\alpha \cdot \beta} - \sqrt{\beta}}{\sqrt{\alpha \cdot \beta} - \sqrt{\alpha}} \cdot \frac{\sqrt{\alpha} + r_i - 1}{\sqrt{\alpha}}. \tag{6}$$

Now, the main process of adding liquidity to Monday Trade Perp can be described. Use ETH-USDC pool as an example:

1. LP chooses α and β and adds a total margin of M USDC.
2. $\frac{M}{P_c \cdot \max\left\{\beta \cdot (1+r_i) - \sqrt{\beta}, \frac{\sqrt{\alpha \cdot \beta} - \sqrt{\beta}}{\sqrt{\alpha \cdot \beta} - \sqrt{\alpha}} \cdot \frac{\sqrt{\alpha} + r_i - 1}{\sqrt{\alpha}}\right\}}$ of ETH long position is created and added into AMM's long position inventory. (An equal amount of ETH-USDC short position is created and added into the LP' s account to balance the market.)
3. This concentrated liquidity will provide pricing from P/α to $\beta \cdot P$.

When P deviates from P_c, the liquidity will have an implied net position calculated by Equation (3). Each concentrated liquidity will always meet the margin requirement for the implied net position within its chosen price range.

When LPs removes their liquidity, the implied net position will be converted into a trading position.

If P goes out of the range of concentrated liquidity, anyone can remove that liquidity and convert it into a trading position, which goes to the account of the original LP.

3.3 Capital Efficiency Boost

The most common definition of capital efficiency is to compare the value of assets required for the same trade size and the slippage. That is equivalent to comparing the M required to result in the same x_{virtual}

in Monday Trade Perp and x in the sAMM, as these two are used in the same constant product formula. For simplicity, we assume $\alpha = \beta$. By Equation (1) and (5), we have

$$\text{CapitalEfficiencyBoost} = \frac{x \cdot P_c \cdot 2}{x_{\text{virtual}} \cdot P_c \cdot (\alpha \cdot (1+r_i) - \sqrt{\alpha}) \cdot \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}}} = \frac{2}{(\sqrt{\alpha} \cdot (1+r_i) - 1) \cdot (\sqrt{\alpha} - 1)} \tag{7}$$

Table 1: Capital Efficiency Boost

α	$1/\alpha,\ \alpha$	Capital Efficiency Boost			
		IMR = 10%	IMR = 5%	IMR = 3%	IMR = 1%
1.01	99.0%, 101.0%	–	–	–	26,666.6x
1.03	98.0%, 103.0%	–	–	2,962.9x	5,364.9x
1.05	95.2%, 105.0%	–	1,066.6x	1,460.9x	2,317.8x
1.10	90.9%, 110.0%	266.6x	404.7x	510.5x	691.0x
1.20	83.3%, 120.0%	102.2x	139.5x	163.3x	196.9x
1.30	76.9%, 130.0%	56.1x	72.4x	81.8x	94.1x
1.50	66.7%, 150.0%	25.6x	31.1x	34.0x	37.5x
2.00	50.0%, 200.0%	8.7x	10.0x	10.6x	11.3x
3.00	33.3%, 300.0%	3.0x	3.3x	3.5x	3.6x
5.00	20.0%, 500.0%	1.1x	1.2x	1.2x	1.3x

When we plug in some α numbers, the following occurs in Table 1. The smart contract restricts the choice of α to be at least $1 + r_i$.

Based on discussions in Sections 3.1 and 3.2, it is trivial to see single sided spot concentrated liquidity falls into the case when $r_i = 100\%$. Indeed, when comparing the publicized boosts from Uniswap v3, we can see similar results.

However, a capital efficiency boost comes with higher risks for LPs, especially in derivatives market settings where leverage further amplifies the risks. These risks can be difficult to grasp with the overlay of leverage and capital efficiency boost. So, beyond the numbers, the mechanism must be clear concerning these risks so that LPs know what they are doing.

Table 2: Capital Efficiency Boost Comparison Across Models

Model	Price Range	Capital Efficiency Boost
Monday Trade Perp	99.99%, 100.01%	39,997.0x
Uniswap v3	99.99%, 100.00%	40,002.5x
Uniswap v3	100.00%, 100.01%	39,998.5x
Uniswap v3	99.99%, 100.01%	20,000.5x

3.4 Implementation Highlights

The smart contract implementation describes concentrated liquidity in a struct called Range. It is stored in the smart contract with an index composed of pair expiry, liquidity owner's address, price range start, and price range end. The struct definition and index information define concentrated liquidity in the smart contract entirely and uniquely.

```
struct Range {
    uint128 liquidity;        // uniswap v3 style liquidity
    uint128 entryFeeIndex;    // snapshot of the global fee index at the time of entry
     uint96 balance;          // margin supplied
    uint160 sqrtEntryPX96;    // AMM price at the time of entry
}
```

To add liquidity, a user sends a transaction on-chain specifying the pair to add liquidity, the price range to provide liquidity, and the margin amount. The smart contract then creates a Range struct for the user with *liquidity* $= \sqrt{k} = \sqrt{x_{\text{virtual}} \cdot y_{\text{virtual}}}$, as calculated by Equation (1), (2).

The smart contract then stores this Range struct in this user's account for this pair with an index composed of price range start and price range end. Based on Equation (3) and (4), the state of a Range struct can be implied using the data stored in the struct, its index, and the current AMM price at any point in time.

4 Fully On-chain Order Book

4.1 Native Irreversible Limit Order

The buy and sell limit orders on single price points are combined with concentrated liquidity to provide liquidity for traders. Unlike AMM, where concentrated liquidity is used as a proxy of limit orders, Monday Trade Perp engine enables native limit orders like those in central limited order book. Orders are placed on price points, not intervals and are irreversible once filled. So makers are certain about the status of their orders. According to Equation (7), limit orders have infinite capital efficiency boost. This is impossible in AMM.

4.2 Matching and Pricing Mechanism

Due to the smart contract limitation, order matching in Monday Trade Perpengine does not follow the first-in-first-out manner in traditional centralized limit order book. The matching process is in Section 5. The characteristics of the matching and pricing mechanism can be summarized below:

1. At a price point where limit orders exist, they are filled before any concentrated liquidity is consumed.
2. Trade volume is allocated to multiple limit orders at the same price proportionally. Just-in-time limit order is welcome.
3. Overall slippage can be greatly reduced when limit orders exist on the AMM price curve.

4.3 Implementation Highlights

In the smart contract implementation, the limit order is defined by a struct Order and is stored in the smart contract with an index consisting of expiry, the owner's address, and price.

```
struct Order {
    uint128 balance;    // margin supplied
    int128 size;        // positive number as buy order, negative number as sell order
}
```

To create a limit order, a user sends a transaction on-chain specifying the pair for this order, the limit price, the order size, and the margin amount. The smart contract then creates an Order struct for the user and stores it in the user's account for this pair with an index composed of the limit price and the nonce of this Order. The nonce is a system-generated number indicating the version of limit orders at this price.

5 Unified Model for Unified liquidity

Monday Trade Perp engine unifies concentrated liquidity and limit orders. In smart contract, Range and Order provide liquidity at each price point. For Range covering a price point, the liquidity, or square root of k, is added for AMM curve-related calculation. For Order at the same price point, their order sizes are added together. In addition, Order is always consumed before liquidity from Range is consumed.

```
struct Pearl {
    uint128 liquidityGross;        // total range liquidity that references this pearl
    int128 liquidityNet;           // amount of net liquidity when the tick is crossed

    uint24 nonce;                  // nonce for record version control
    int96 left;                    // orders to be taken
    int96 taken;                   // orders to be filled

    uint128 fee;                   // should only be distributed w.r.t. filled order
    uint128 entrySocialLossIndex;  // social loss index borne by taken but unfilled order

    int128 entryFundingIndex;      // funding index owned by taken but unfilled order
}
```

The collection of concentrated liquidity covering a price point and all open limit orders on the same price point is described in a struct called Pearl and is stored in the smart contract indexed by price. Monday Trade Perp engine manages the collection of Pearl along with the AMM price curve. Among all the Pearl fields, liquidityGross and liquidityNet are related to Range and the rest are related to Order.

Native irreversible limit orders in Monday Trade Perp engine eliminate the necessity to create concentrated liquidity with extremely narrow range to mimic limit order. Consequently, the fee distribution mechanism in Monday Trade Perp engine is significantly simplified compared to other spot-concentrated liquidity DEXes. This drastically reduces engineering complexity.

In fact, Pearl serves as the pool for limit orders, which is key to realizing the asynchronous design of limit orders. In this way, logic on the takers' side is greatly simplified, where takers take as much as they want and worry nothing else. With the fungible approach of Pearl, the gas cost of a trade is directly proportional to the price impact, i.e., the number of ticks crossed.

For a trade of size S_0, the process of trading, or consuming unified liquidity, follows the steps below and is shown in Figure 2.

1. In the Pearl of current price P_0, check if there are unfilled limit orders.
   (a) If not, go to step 2 with $S_1 = S_0$.
   (b) If so, fill the limit orders as much as possible.
      i. If S_0 is fully filled, terminate. (Note that the current price does not change in this case.)
      ii. If not, continue to step 2 with the remaining size S_1.
2. Find the Pearl at the next price P_1.
   (a) Trade for size S_1 on the AMM curve between Pearl at P_0 and P_1.
      i. If S_1 is fully filled, terminate. (Note that the current price does change in this case.)
      ii. If not, update the current price as P_1 and go to step 1 with the remaining size S_2.

6 Stabilization Mechanism for User Protection

6.1 Prices

Monday Trade Perp maintains four prices for different purposes.

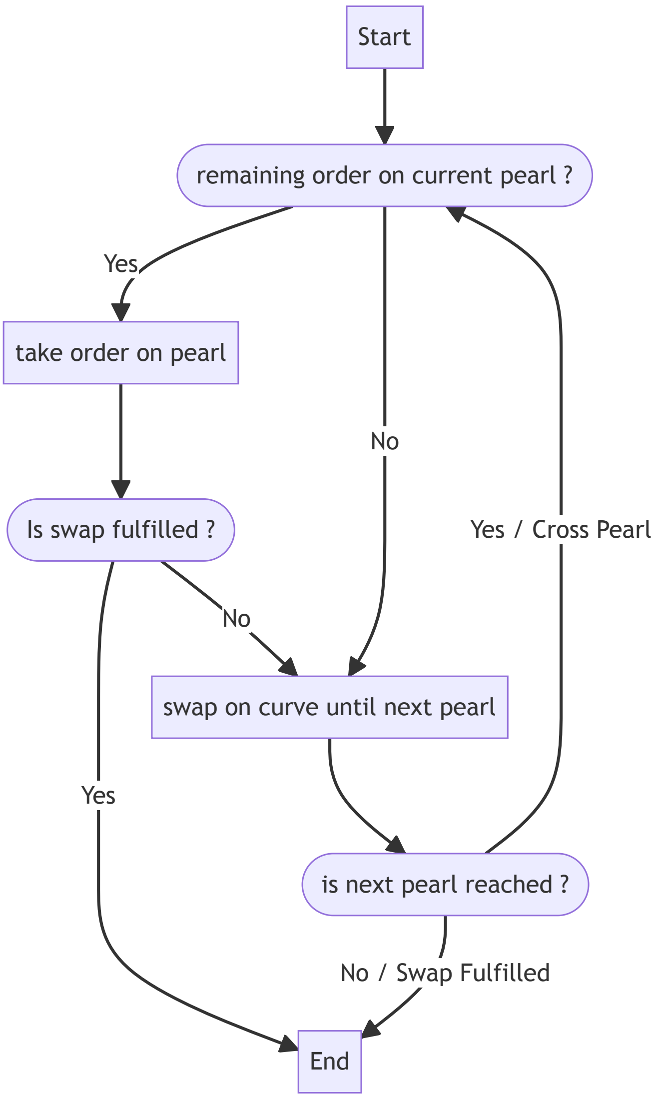

Figure 2: Overview of Swap Procedure

- P_{fair}: Price quoted by the Monday Trade Perp engine by combining all the concentrated liquidity and limit orders, like the mid-price of a CLOB system. This price is used to determine if concentrated liquidity is out of its price range. This price is also used when a concentrated liquidity is converted into a trading position. In summary, this price is used for all liquidity-related calculations.
- P_{mark}: Price used to determine the unrealized P&L and margin requirement, including the initial margin requirement to open a position. In summary, this is the price for position risk management.
- P_{spot}: Price aggregated from oracles such as Chainlink and various spot DEXes.
- $P_{\text{settlement}}$: Time weight average price from oracles used for final settlement of positions in a dated futures market.

Just as the derivative markets are related to the spot markets but function independently, these prices can coincide but are often different. P_{fair} solely depends on the trading and liquidity within the pool, while P_{spot} and $P_{\text{settlement}}$ solely depend on spot market data and time. P_{mark} is based on P_{spot} but also incorporates a daily interest component of the underlying trading pair.

6.2 Continuous Funding Fee

As perpetual futures never expire, to guarantee the convergence to the spot market index price, Monday Trade Perp engine employs continuous funding for its perpetual futures markets. The principle is to expect the deviation to converge in a daily cycle, and thus, the funding fee ratio for a period is defined below:

$$\mathsf{FundingFeeRate} = (P_{\text{fair}} - P_{\text{spot}}) / P_{\text{spot}} \cdot \Delta t / 86400$$

Δt is the time difference in seconds between the current timestamp and the last timestamp when the funding fee rate is calculated. Like other centralized and decentralized perpetual futures markets, the funding fee encourages traders to return the perpetual futures price to the index price by continuously rewarding positions against such deviation and punishing positions in favor of such deviation.

6.3 Liquidations

When the margin of a trading position falls below the maintenance margin requirement, that position is subject to liquidation. Liquidation in Monday Trade Perp engine is performed primarily via the take-over approach, where the liquidator takes over the target trading position along with the remaining margins and tops up the margin to meet the initial margin requirement.

Effectively, the remaining margin of the trading position to be taken is a potential profit for liquidators. In addition, Monday Trade Perp provides a liquidation mechanism where trading positions below the maintenance margin requirement are forced to trade against the AMM directly to close the position. Both approaches support partial liquidation.

DISCLAIMER

This document serves solely for informational purposes. It is not intended as investment guidance, a suggestion, or an invitation to engage in the buying or selling of any products or investments. It should not be employed for assessing the appropriateness of any investment choices, nor should it be considered a source of accounting, legal, or tax counsel, or investment suggestions. The views expressed in this document are those of the authors at the time of publication and are not made on behalf of Monday Trade. They may not necessarily represent the perspectives of Monday Trade or individuals associated with the project. The expressed opinions are subject to change without notice or updates.