Fusion AMM (FAMM)

nftperp v2 introduces a hybrid liquidity protocol that combines constant liquidity AMM with decentralized limit orderbook (DLOB).

Every positions taken against the AMM and/ or DLOB as a direct counterparty on the other side, which fixes the counterparty issue nftperp v1 faced.

We call this - Fusion AMM.

The benefits of having an AMM as the base source layer of liquidity are:

There’s always liquidity available for takers to enter or exit positions

Isolated and customizable pool parameters for individual market risks

Easy bootstrap for new pair

Allows anyone to be a maker on the protocol with automated bid/ ask strategy

Combining with nftperp’s DLOB, the Fusion AMM inherits the benefits of limit orderbook liquidity:

Independent and ease of price discovery

Scaling - increase market depth and reduce price impact on significant orders

Direct counter parties for takers and makers

Minimal price impact on markets when order conditions are matched

Less slippage and better pricing around the mid-price.

Keytakeaway : The Fusion AMM is creating an hybrid model using both competitive advantage from each system, increasing market depth around the market price and reducing price impact on large orders via the DLOB and using the AMM as an independant market makers inside the DLOB and is providing liquidity on both tails of the curve.

Fusion AMM Matching Logic

Limit Orders automatically go to the orderbook. However, Market Orders might be executed in either the limit order book, the AMM, or both. The protocol will pick a path depending on the size of the market order and liquidity available on the orderbook.

Let’s observe how price/ slippage functions in two environments:

a) AMM liquidity only:

AMM: BAYC/ ETH

BAYC (x) reserve: 100

ETH (y) reserve: 5000

AMM BAYC price: 50 ETH

K value: 500000 (x*y)

Alice places a long position on BAYC with a notional value of 100 ETH.

Based on the bonding curve, Alice would receive roughly 1.96078 BAYC with 100 ETH market buy with AMM external LPs being the counterparty.

The price of BAYC after Alice’s market buy order would now be 52.02 ETH (4.04% increase)

Post Alice trade AMM state:

BAYC (x) reserve: 98.03921569

ETH (y) reserve: 5100

AMM BAYC price: 52.02 ETH

K value: 500000 (x*y)

b) AMM liquidity + orderbook liquidity:

AMM: BAYC/ ETH

BAYC (x) reserve: 100

ETH (y) reserve: 5000

AMM BAYC price: 50 ETH

K value: 500000 (x*y)

limit sell order of 1.5 BAYC at the price of 50.5 from Bob

Alice places a long position on BAYC with a notional value of 100 ETH.

The 100 ETH would be broken down to roughly 24.93 ETH being filled by the AMM, pushing the BAYC price up to 50.4999 and the rest 75.07 ETH filled by Bob's 1.5 BAYC limit sell order.

Given Bob's limit sell order is priced at 50.5 ETH per BAYC, the notional value of the order would have 75.75 ETH available to be filled by Alice's 75.07 ETH.

The price of BAYC after Alice's market buy order would now be 50.5 (0.99% increase)

In summary, Alice's 100 ETH market order resulted in 0.49612 BAYC from AMM + 1.48653 BAYC from Bob's limit sell order.

Alice's BAYC long notional value: 1.98265 BAYC

Bob's BAYC short notional value: 1.48653 BAYC

Post Alice + Bob trades AMM state:

BAYC (x) reserve: 99.50387369

ETH (y) reserve: 5024.93

AMM BAYC price: 50.5 ETH

K value: 500000 (x*y)

AMM + orderbook liquidity dramatically improves user experience as Alice receive more BAYC from her market buy and price slippage reduced by nearly 75%.

Fully On-Chain Limit Orders

To make a limit order, a trader must make an onchain deposit. Once deposited, a user can create orders as long as they have enough margin deposited — or they can set a flag to take their margin atomically at the time order creation. This order is stored onchain and matched with incoming market orders.

On-Chain Conditional Orders w/ Off-Chain Execution

Conditional Orders (Take-Profit & Stop-Loss) are submitted on-chain and executed by keepers (similar to liquidations). To incentivize timely execution, the keeper fee is increased the closer to the mark price is to the trigger price of the order. Once executed, the conditional order is treated like a market order and matched with the orderbook & amm.

Order Types (Limit, Take-Profit, Stop-Loss)

NFTPerp V2 supports all standard order types

1) Limit Order: Users can add basic limit orders on either side of the orderbook. Orders are matched on-chain with incoming market orders

2) Stop-Loss & Take-Profit Orders: These reduce-only orders are executed w/ intent to either prevent too much loss (executed at or below trigger from a long position) or take a profit (executed at or above trigger from a long position).

NLP - an AMM external LP model

We have yet to finalize the design choice for the Liquidity Provision. But nftperp V2 supports Liquidity Provision. We will choose one of the following mechanisms (or a mix of both) for liquidity provision:

NLP - Adding Liquidity to the AMM: Adding Liquidity to the AMM will reduce slippage and automate the discovery of K. In this design, LPs will take the reverse position compared to the traders. They will be exposed to a form of IL and in return receive funding (most of the time) and protocol fees.

NDLP Adding Liquidity to the Limit Orderbook: An LPs liquidity is automatically put in both sides of the order book. In this

design, LPs will be exposed to a form of IL and receive funding and the protocol fee.

Currently we are discussing using protocols such as Elixer so anyone can utilize automated market making strategies by depositing funds into an LP pool that execute MMs on the DLOB.

Oracle

In nftperp V1, we used a concept called True Floor Price. True Floor Price was the TWAP of the trade data of a NFT after filtering outliers using statistical methodologies and volatility scoring. This average is often different from the floor price. And we observed that people prefer an Oracle Price they understand better. So in nftperp V2, we have collaborated with MetaQuants to use a different floor price mechanism.

The new mechanism averages the TWAP of the best bid and best ask across different markets after removing the outliers. Then it compares itself with the trade price TWAP (the V1 Oracle) to remove bias. We found that this mechanism integrates movements fast and is very reliable.

Funding Payments

Funding for nftperp follows a standard per hour funding period model.

Funding occurs every 1 hour, 24 hours a day. Traders only pay or receive funding if they hold a position at one of these times. Funding payments are needed to prevent the price of perp from deviating too far from the underlying assets. These are payments made between traders depending on whether they hold long or short positions. Funding consists of regular payments between buyers and sellers, according to the current funding rate. When the funding rate is above zero (positive), traders that are long (contract buyers) have to pay the ones that are short (contract sellers). In contrast, a negative funding rate means that short positions pay longs.

nftperp v2's funding payment uses a symmetric funding rate where longs and shorts have the same rate, given the long short ratio will always be 50/50.

Liquidations (to be updated)

When a trader opens a leveraged position, they use their collateral to borrow capital from the protocol to buy or sell an asset.

When the value of that trader's position moves against them, their loss will move towards their margin i.e., initial collateral. The protocol is now put at risk where a sudden price movement could turn the trader's position worth less than the initial collateral. When the value of the trader's position is getting too close to the value of the initial collateral, the protocol will liquidate that position in order to maintain its solvency.

For example, if a trader opens a 5x leverage position using 1 ETH, their position would have a notional value of 5 ETH, of which 4 ETH were borrowed (20% initial margin ratio).

The protocol's smart contract requires a minimum ratio between the notional value and the margin — the minimum maintenance margin .

When the trader's margin ratio becomes lower than the minimum maintenance margin ratio, liquidation will be triggered by keeper bots.

$$MarginRatio = (Initialmargin + UnrealizedPnL) / OpenPositionNotionalSize$$

$$UnrealizedPnL = (ExitPrice - EntryPrice) * Positionsize + FundingFee$$

$$PositionNotional = PositionSize * MarketPrice$$

$$OpenPositionNotionalSize = PositionSize * EntryPrice$$

Keeper bots trigger liquidations when a position is liquidatable at the current index price and the mark price (note: a position can only be liquidated when both mark price and index price puts it under minimum maintenance margin). They earn 20% of the remaining notional as a reward for providing this service. The remaining margin will be deposited into the Insurance Fund.

(We'll open source the keeper bot post public mainnet launch, so anyone can call the liquidation function and earn the rewards.)

Trading Fees

Every trade on nftperp has a 0.3% taker fee based on the position size.

The 0.3% is then broken down to:

0.15% goes to nftperp NLPs

0.15% goes to nftperp insurance pool

Insurance Pool (LP staked)

The nftperp insurance pool is similar with its insurance fund, but it is funded by liquidity providers. The purpose of the insurance pool is to cover protocol bad debt (same as insurance fund). In exchange, the insurance pool shares incoming revenue (trading fees, liquidation penalties etc).

0x11Cce125f8A6219cde75C90e1fD85Ba2D87e3FfB

Untitled