Trader Joe

JOE v2.1
Liquidity Book

MountainFarmer
mountainfarmer@traderjoexyz.com

Louis
louis@traderjoexyz.com

Hanzo
hanzo@traderjoexyz.com

Wawa
wawa@traderjoexyz.com

Murloc
murloc@traderjoexyz.com

Fish
fish@traderjoexyz.com

Abstract:

We introduce Liquidity Book (LB), a novel protocol for structuring liquidity in a decentralized exchange. The LB protocol enables the creation of unique and dynamic liquidity structures for a paired asset market. Liquidity is pooled in constant price bins which are aggregated to establish a market.

November 2022

1 Introduction

Trader Joe (v1) was launched in 2021 as an AMM with Uniswap v2 liquidity contracts as its backbone [1]. As DeFi advances, we have concluded that capital efficiency and liquidity flexibility are urgent issues that demand protocol development. In this paper, we present our design for the next stage of the JOE decentralized exchange.

1.1 Concentrated Liquidity

Tick bound liquidity, as implemented in Uniswap v3 (v3), pioneered the concept of concentrated liquidity for a general automated market maker (AMM) [2]. In Uniswap's implementation, individual liquidity positions are established which provide reserves over a restricted price range. The reserves of each position are exchanged over its price range following a common constant function formula. This common function allows individual position reserves to be aggregated together to form the market.

Even before the concentrated liquidity term was coined, StableSwap/Curve provided a similar solution for stable exchange rate reserves. Curve v1 utilizes an invariant curve with an implied static peg [3]. While the liquidity is unbounded, the bulk of reserves concentrate liquidity around the peg price. Curve v2 introduces a more general solution using an internal price oracle to upgrade to a dynamic peg [4].

Constant function algorithms used in Uniswap and Curve restrict the ability of liquidity providers to deploy tailored market making strategies. We propose a new protocol with Discretized Concentrated Liquidity to provide a flexible infrastructure layer for market participants. Discretized Concentrated Liquidity has been previously presented by the iZiSwap limit order implementation [5].

1.2 Liquidity Book

The Liquidity Book (LB) arranges liquidity of an asset pair into discrete price bins. Reserves that are deposited into a liquidity bin are available to be exchanged at the constant exchange rate (price) that is defined for the bin. The market for the asset pair is constructed by aggregating all of the discrete liquidity bins. For example, a market liquidity structure supplied by different market participants (color coded) could be represented by the figure below.

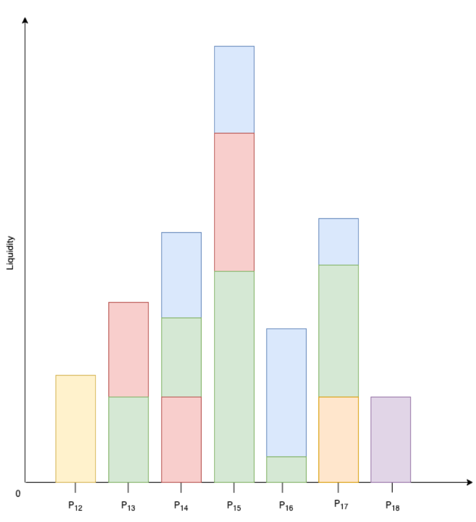

Figure 1: Example Liquidity Structure

2 Protocol Design

2.1 Reserve Pricing

Following AMM convention, price (*P*) is defined as the rate of change in *Y* reserves per change in *X* reserves. Price represents the value of the base asset (*X*) denominated in the quote asset (*Y*) terms.

$$P = \frac{\Delta y}{\Delta x} \tag{2.1}$$

2.2 Bin Liquidity

Liquidity (*L*) is defined to be the nominal value of reserves deposited into a discrete bin and is denominated in the quote asset. Liquidity is conserved as the bin's reserve composition changes and can be described in Eq 2.2 as the value sum of its composite assets.

$$P_i \cdot x + y = L \tag{2.2}$$

Each individual bin is a constant sum market with its own unique bonding curve. The bin's curve is fully defined by the price parameter of the bin (static) and the total liquidity reserves deposited into the bin (dynamic). As shown in Figure 2, the *y* reserve intercept represents the bin's total liquidity (*L*) while slope is defined by the bin's price (P_i).

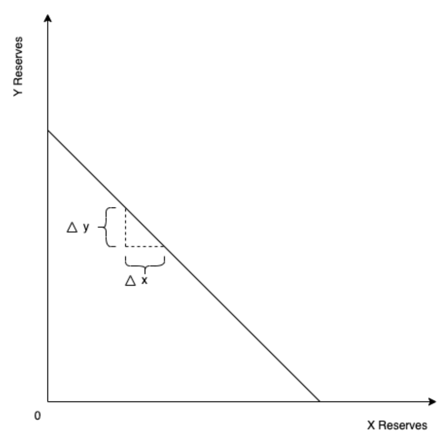

Figure 2: Reserve Bonding Curve

2.3 Bin Composition

Due to the design above, the composition of reserves in a liquidity bin must be independent of both price and liquidity. An additional variable can be used to describe the reserves available in a bin. This variable, composition factor (*c*), will be defined as the percentage of bin liquidity currently held as *Y* reserves. Composition (c) is bounded by [0,1].

$$c \equiv \frac{y}{L} \tag{2.3}$$

Asset reserves of each liquidity bin can be fully described by bin liquidity, *L*, and composition, *c*, as shown in Eq 2.4 and 2.5.

$$y = cL \tag{2.4}$$

$$x = \frac{L}{P_i}(1 - c) \tag{2.5}$$

An example composition of *X* and *Y* reserves in a liquidity bin can be visualized as shown in Figure 3.

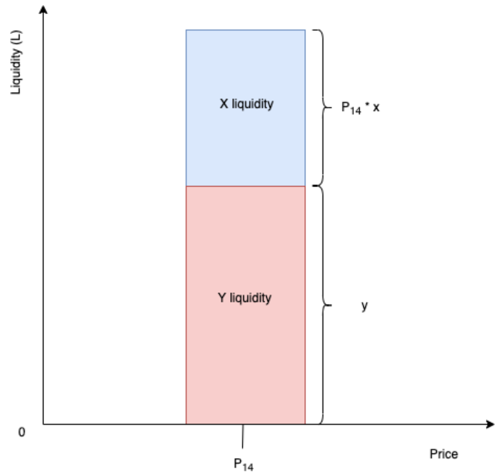

Figure 3: Bin Reserve Composition (P_{14} bin)

2.4 Market Aggregation

Aggregation of individual liquidity bins for an asset pair establishes the market for that pair. The reserve state of liquidity bins is expected to transition between *X* reserves and *Y* reserves as market pricing changes. Liquidity bins that are priced below market will contain only asset *Y* and liquidity bins priced above market will contain only asset *X*. Market price is therefore determined to be the lowest price bin containing *X* reserves.

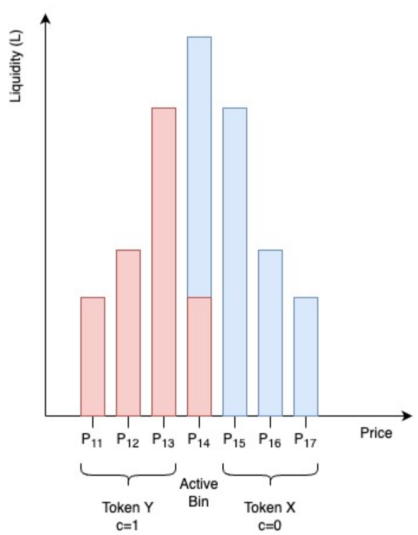

Figure 4: Aggregate Liquidity Composition

2.5 Market Making

With discrete liquidity bins, liquidity providers are able to create custom liquidity structures that allow them to easily manage their positions according to their individual objectives, projections, and risk profiles. A liquidity provider can concentrate liquidity around certain price points or spreads and adjust individual liquidity bins without impacting their liquidity positioning in other bins. Overall, the liquidity book design allows for the fine tuned range and depth adjustments that market makers require to operate successfully.

3 Protocol Architecture

3.1 Market Definition

Contractually, the price range of every market is constrained to (2^{-128},2^{128}). This price range is discretized into bins according to a discretization parameter called the bin step (*s*). The smaller the bin step the greater the quantity of bins in the market. A unique market is defined by the (*X, Y, s*) set.

The total number of liquidity bins in a given market can be described as $2*N_b$, where N_b is calculated using Eq 3.1. N_b represents the maximum number of bins that can exist between price parity (par) and the max price in a market with a given bin step.

(3.1) $$N_b = argmax_i\left((1+s)^i < 2^{128}\right)$$

In addition to the price constraint, the maximum quantity of bins is limited to an amount less than 2^{24}. Assigning half of the potential bins to a value less than par and half to a value greater than par in a market with the maximum number of bins results in a par bin identifier of 2^{23}, which is assigned to the constant b in Eq 3.2. In all markets, bins will be assigned a positive integer identifier (*i*) that ranges between [$b - N_b$, $b + N_b$].

(3.2) $$b \equiv 2^{23}$$

3.2 Bin Price

The price prescribed to each bin is a function of the pair's bin step (*s*) and the bin's identifier (*i*), as shown in Eq 3.2. The bin step parameter determines the constant percentage increase or decrease in price between each incremental bin.

(3.3) $$P_i = (1+s)^{i-b}$$

3.3 Liquidity Tracking

To optimize liquidity tracking, bins are indexed in a tree that is created by nesting three 256 bit arrays as depicted in Figure 5. In this tree, each bin is assigned a location path through the nested arrays. When a bin contains liquidity, a non-zero value is assigned to the three array elements that represent the bin's location. With this tree, the liquidity state of the market can be efficiently searched during swapping and/or monitored externally.

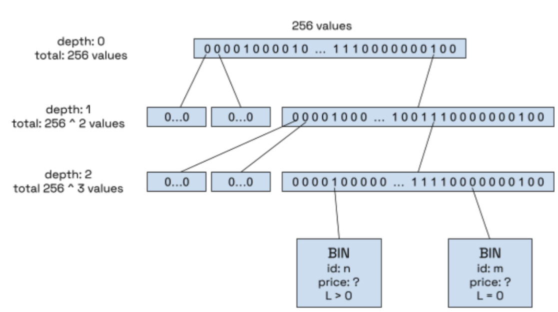

Figure 5: Bin Index Tree

3.4 Trade Routing

The LB router will provide added security and slippage checks for swaps and liquidity deposits/withdrawals. Swaps will also be routed through legacy AMM pairs when better pricing is found.

3.5 Liquidity Tokens

A new liquidity token standard has been developed specifically for the LB protocol to support the large quantity of bins in each market. By design, liquidity (*L*) in an LB market is composable across bins since it is commonly valued in *Y* terms and is independent of the state of the bin. Therefore, the liquidity receipt token, *LBToken*, will carry an equal liquidity value irrespective of the bin it is contained in. This property enables bin liquidity to be seamlessly bundled together across bins.

LBTokens will essentially resemble legacy ERC-20 liquidity tokens with the following key difference: they are assigned an id when minted that matches the bin identifier (*i*) where the liquidity resides. The LBToken contract follows the ERC-1155 multi-token standard except that it is restricted to containing fungible ERC-20 tokens.

LBTokens are fungible, price-located liquidity receipt tokens.

3.6 Market Parameters

Each market will have a set of parameters that govern the fee rates for the market, further detailed in section 4.3. Initial parameter values will be assigned according to the bin step parameter selected for the market. Markets may only be established with bin steps that have associated fee parameters in the factory. Fee parameters may be adjusted after initialization.

Parameter	**Symbol**	**Bounds**
binStep	s	[0.0001, 0.01]
baseFactor	B	[0, 6.5535]
filterPeriod	t_f	[0, 65535] (seconds)
decayPeriod	t_d	[0, 65535] (seconds)
reductionFactor	R	[0, 1]
variableFeeControl	A	[0, 1677.7215]
protocolShare	ϕ	[0, 0.25]
maxAccumulator	v_{max}	[0, 1677.7215]
sampleLifetime	q	[0, 65535] (seconds)

3.7 Oracle

LB markets will have an oracle that records the following values to the blockchain:

- *timestamp*
- *cumulativeID*
- *cumulativeAccumulator*
- *cumulativeBinCrossed*

Cumulative values are useful for determining changes over variable amounts of time. The oracle supplies market price and volatility/fee metrics as it is useful information for algorithmic trading and liquidity provision. The *sampleLifetime* setting is a market parameter for the sampling rate that is defined when the market is created. Oracle sample size may be increased by users when needed.

4 User Interactions

4.1 Add/Remove Liquidity

Adding and removing liquidity to/from an active liquidity bin will conserve the price (*P*) and composition (*c*). For a given liquidity adjustment, the required quantity of reserve *X* and reserve *Y* can be calculated using Eq 4.1 & 4.2 below.

(4.1) $$\Delta L = \frac{P}{(1-c)}\Delta x$$

(4.2) $$\Delta L = \frac{1}{c}\Delta y$$

For all 'inactive' liquidity bins, liquidity can be added exclusively in reserve *X* or reserve *Y* as determined by the composition bounds.

(4.3) $$\Delta L_{c=0} = P \cdot \Delta x$$

(4.4) $$\Delta L_{c=1} = \Delta y$$

Liquidity added to the active bin will automatically be swapped if it does not match the bin's composition and will incur a fee (Section 4.4). Transactions that attempt to add liquidity to inactive bins and are not consistent with the bin's composition will revert.

4.2 Swaps

Exchanging reserves within a liquidity bin will conserve liquidity and price, therefore only reserve composition will change. Reserves are exchanged following Eq 4.5 while the composition remains in its bounds of [0,1].

(4.5) $$\Delta y = - P\Delta x$$

The reserves that are available in the liquidity bin until the composition bounds are reached can be calculated following Eq 4.6 and 4.7.

(4.6) $$\Delta c = - \frac{P}{L}\Delta x$$

(4.7) $$\Delta c = \frac{1}{L}\Delta y$$

When a swap requires more liquidity than is available in the current bin, the swap will exhaust the liquidity in each consecutive bin before moving to the next adjacent bin with liquidity.

4.3 Swap Fees

Fees will be collected by the protocol to compensate liquidity providers for the trading activity that occurs. The total swap fee (f_s) will have two components, a base fee (f_b) and a variable fee (f_v) that is a function of instantaneous price volatility. The fee rate will be applied to the swap amount in each liquidity bin and distributed proportionally to the liquidity providers in that bin following a distribution to the protocol. Fees will be held separate from liquidity and claimable by liquidity providers. The total fee for a swap crossing n bins can be calculated following Eq 4.8.

(4.8) $$f_s = \sum_{k=0}^{n}\left[(swap\ amount)_k \cdot (f_b + f_v)_k\right]$$

4.3.1 Base Fee

The base fee rate (f_b) is a function of the bin step (s) and is scaled by a *baseFactor* (B), as shown in Eq 4.9. The base fee represents the minimum rate for all swaps.

(4.9) $$f_b = B \cdot s$$

4.3.2 Variable Fee

The variable fee (f_v) is designed to compensate liquidity providers for instantaneous volatility and incentivize liquidity providers to actively manage liquidity around a moving price. The variable fee is calculated per bin (k) using Eq 4.10.

(4.10) $$f_v(k) = A\,(v_a \cdot s)^2$$

Instantaneous volatility is captured in a newly introduced volatility accumulator variable, v_a. A is the *variableFeeControl* market parameter, which is used to scale the variable fee component depending on the expected dynamics of the market.

4.3.3 Volatility Accumulator

Volatility on LB is derived from the number of bin changes that occur over time. Each bin change represents a fixed price movement that is defined by the bin step. An accumulator is implemented to extend the application of bin crossovers as a measure of volatility beyond a single transaction. The volatility accumulator (v_a) is defined as a function of the bin crossovers within a single transaction (k), a reference bin index (i_r), the active bin at the beginning of the transaction ($i_{k=0}$), and a reference volatility value (v_r) as shown in Eq 4.11.

(4.11) $$v_a = v_r + \left|i_r - (i_{k=0} \pm k)\right|$$

Where the reference values are calculated at the beginning of each transaction and depend on time elapsed since the last transaction (t):

(4.12) $$v_r = \begin{cases} v_r & t < t_f \\ R \cdot v_a & t_f \le t \le t_d \\ 0 & t_d \le t \end{cases}$$

(4.13) $$i_r = \begin{cases} i_r & t < t_f \\ i_{k=0} & t_f \le t \end{cases}$$

Therefore, the volatility accumulator measures the bin distance from the bin where the swap is occurring to a reference bin and adds a carryover volatility (v_r) to that distance. The reference bin updates periodically to the current bin when there are no transactions in the market for a period exceeding the *filterPeriod* (t_f). The reference volatility also periodically updates based on transaction frequency in the market. Reference volatility updates to the current volatility decayed by the *reductionFactor* R when t is greater than the *filterPeriod* (t_f), or it completely resets to 0 when t is greater than the *decayPeriod* (t_d).

To prevent excessive and persistent fees, each market will limit the maximum value of the accumulator with the *maxAccumulator* parameter, v_{max}.

4.4 Active Bin Composition Fee

Liquidity that is added to the active bin will automatically be swapped across reserves if the composition is not equal to the bin's composition. The amount swapped is calculated so that the resulting assets in the bin equal those that would result if the user were to swap prior to adding liquidity. The amount swapped incurs a fee that approximates the current market swap fee.

(4.14) $$f_c = f_{s,k} \cdot \left(1 + \frac{f_{s,k}}{(swap\ amount)}\right)$$

4.5 Protocol Fees

A portion of swap fees may be retained by the protocol (f_p) and will be controlled by the *protocolShare* (ϕ) parameter. Each individual market will have its own *protocolShare* parameter to govern fee distributions for that specific market*.* After protocol fees are collected, the remainder will be sent to liquidity providers.

(4.15) $$f_p = \phi \cdot f_s$$

5 DEX Improvements

5.1 Impermanent Loss

Variable fees provide liquidity providers compensation for managing impermanent loss under various trading dynamics. Given impermanent loss occurs due to price volatility, the volatility accumulator is a natural mechanism to adjust fees to market dynamics. Further, liquidity providers have opportunities to outperform this expected payoff due to the time decay of the volatility accumulator. This payoff is shown in Figure 6. Since the accumulator is based on bin steps, a tracking error occurs under extremely volatile events due to large deviations in nominal price changes.

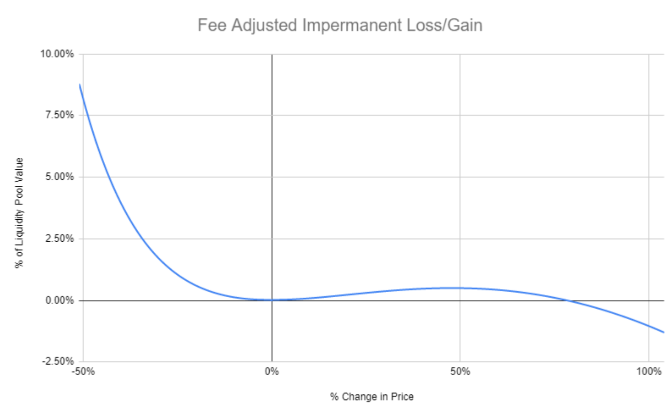

Figure 6: Fee Adjusted Impermanent Loss/Gain

5.2 Liquidity Depth

Improving the capital efficiency of the exchange increases market depth for a given amount of capital. Capital efficiency can be defined as the reduction in capital required to provide liquidity equivalent to v1. For LB, comparable liquidity is found by equating the price impact from a swap on v1 to a single bin step. The efficiency equation resulting from that definition is shown below with efficiency limit results tabled in Figure 7. These limits represent the maximum efficiency for each market with a given bin step.

(5.1) $$\varepsilon = \frac{-\sqrt{1+s}}{1-\sqrt{1+s}}$$

Bin Step	Basis Points	Efficiency	Bin Step	Basis Points	Efficiency
0.1	1000	21.5x	0.005	50	401.5x
0.05	500	41.5x	0.0025	25	801.5x
0.02	200	101.5x	0.001	10	2001.5x
0.01	100	201.5x	0.0005	5	4001.5x
0.0075	75	268.2x	0.0001	1	20001.5x

Figure 7: Capital Efficiency Limits

As seen in Figure 7, LB's efficiency is comparable to Uniswap v3 (v3). An evenly distributed LB position can be used to approximate a v3 position. If liquidity were to be distributed equally across bins in *X* tokens, then the relative difference in swap value between LB and v3 across the position is described by Eq 5.2. This equation can be plotted for a set of positions with various price ranges discretized into different bin steps, as is shown in Figure 8.

(5.2) $$(v3\ diff)_{xdist} = \frac{1}{ns}\left(\frac{(1+s)^{n}-1}{\sqrt{(1+s)^{n-1}}}\right)$$

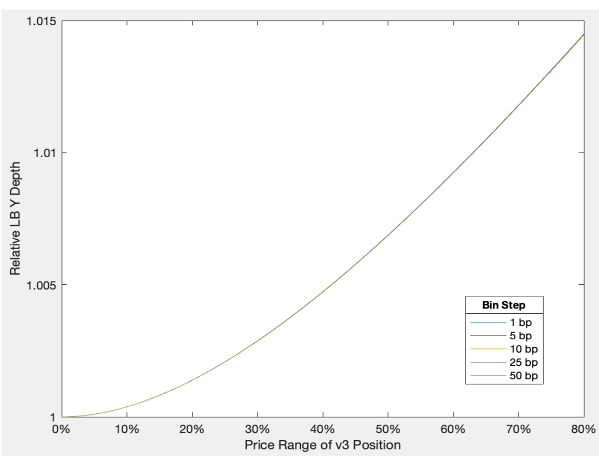

Figure 8: Market Depth Comparison to Uni v3

As evident in Figure 8, position price range is a substantially stronger factor for relative market depth than the bin structure. In fact, for a position spanning at least 2 bins the difference in depth across step selections is negligible. With the distribution specified above, the market depth of LB is within 1% of Uniswap v3 for any position that spans less than 60% price range. Moreover, the difference in depth is due almost entirely to the choice of liquidity distribution across the range (as compared to v3's geometric sequence) and is only significant over large range positions. These findings only further illustrate the value of the flexible liquidity distribution that is enabled by the LB protocol.

6 Conclusion

Liquidity Book (LB) is a novel design for structuring the liquidity of a decentralized exchange. It allows liquidity to be discretized into fixed price bins, improving slippage and swap pricing. Unlike prior concentrated liquidity protocols, LB avoids high impermanent loss to liquidity providers. LB liquidity structures allow for further composability and we are keen to explore new use cases with the DeFi community.

References

1. H. Adams, N. Zinsmeister, and D. Robinson. 2020. *Uniswap v2 Core*. Retrieved Jun 21, 2022 from https://uniswap.org/whitepaper.pdf
2. H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson. 2021. *Uniswap v3 Core*. Retrieved Jun 21, 2022 from https://uniswap.org/whitepaper-v3.pdf
3. M. Egorov. 2019. *StableSwap - efficient mechanism for Stablecoin liquidity*. Retrieved Jun 21, 2022 from https://curve.fi/files/stableswap-paper.pdf
4. M. Egorov. 2021. *Automatic market-making with dynamic peg*. Retrieved Jun 21, 2022 from https://curve.fi/files/crypto-pools-paper.pdf
5. J. Yin and M. Ren. 2021. *iZiSwap: Building Decentralized Exchange with Discretized Concentrated Liquidity and Limit Order*. Retrieved Jun 21, 2022 from https://izumi.finance/paper/dswap.pdf

Revisions

November 2022

3.6: Updates to parameter allotments and limits.
4.3.1: Increased upper limit on base fee to permit larger parameter values
4.3.3: Volatility accumulator redesigned. Added stored bin and volatility reference variables to enable volatility to net in high frequency conditions.
4.4: Added to highlight active bin composition fee
4.5: Moved protocol fees from 4.3.4 since they apply to all fees collected
Multiple: Finalized variable names & symbols