

Liquidity Module V1 : Mechanism Explained

1. Context

a. Utility of the Cosmos Hub

- The utility of the Cosmos Hub is to provide core functionalities related to the soon-available IBC(Inter-Blockchain Communication) protocol
- The most important IBC-related functionality is the trading functionality among transferred coins into the Cosmos Hub
- The trading functionality has to safely store and manage a significant amount of external assets to facilitate the utility. Therefore, it is necessary for the Cosmos Hub, who has the largest dPoS security, to take the role to provide this important utility for a broad range of potential users

b. AMM(Automated Market Makers)

- Many DeXs were introduced to the cryptocurrency ecosystem for the past several years, but most of them failed to attract a significant user-base because it was very difficult to provide sustainable high-quality liquidity to the exchanges.
- Difficulties are caused by
 - 1) Expensive cost to hire professional market makers, because they require highly-trained financial engineers and significant capital investment
 - 2) Short-term profit oriented and high-return low-risk nature of market makers, which result in unstable and low quality liquidity
 - 3) Unnecessary large number of onchain transactions for limit orders and cancel orders
- Now, AMM model is introduced which solves most of the problems of traditional orderbook-based DeXs
 - 1) It lowers the barrier of market maker participation to anyone who does not have financial engineering skills nor significant amount of capital, which results in more efficient competition among market makers, hence cheaper liquidity cost for the exchanges
 - 2) The mechanism of AMM forces any market makers of AMM to provide liquidity in a pre-defined algorithmic way, so that the provided liquidity becomes having higher quality and better stability
 - 3) AMM drastically reduces the necessary number of transactions for its functionality, which is more adequate for decentralized environment

c. Interchain AMM as a Vision of the Cosmos Hub

- Most Ethereum-based AMMs are only dealing with ETH and ERC20 coins
- From Binance Exchange, in the top 20 coins, the volume of all ETH and ERC20 coins are occupying only 25%. It means that the Cosmos Hub AMM has potential user base of 3 times more than the potential user base of Ethereum-based AMMs, by connecting different kinds of blockchains via interchain communication technologies
- Therefore, the Cosmos Hub naturally has its vision to become most inclusive multi-chain AMM utility provider

d. B-Harvest is building Liquidity Module for the Cosmos Hub

- B-Harvest is one of the strongest validators and contributors of the Cosmos ecosystem
- B-Harvest has very skilled Cosmos-SDK engineers who have various experience to research and develop utilities with Cosmos-SDK
- B-Harvest has strong knowledge/experience background on traditional finance, including stock/derivatives exchanges, market making, algorithmic trading, arbitrage trading, low-latency trading, and asset risk management
- Above characteristics of B-Harvest makes the team the best talents for building AMM for the Cosmos Hub. So the Tendermint team decided to fund B-Harvest to design and implement the production level Liquidity Module, which will serve AMM functionality on the Cosmos Hub.

2. General Philosophy of the Liquidity Module

a. Stick to the Basic

- The basic design of Liquidity module will follow the successful industry design standard of AMM, especially liquidity pool and constant product equation methodology from many current successful AMMs
- Too unique design of AMM can cause unnecessary design risk which results in failure of mass adoption

b. Feature Extensibility and Continuous Evolution

- We think that every new technology is a dynamically evolving creature
- Rather than having too specified, narrowed design from the beginning, the Liquidity module has most common and standard design with capability to expand its features in future
- The module will be continuously evolving to mitigate market demands, to have better incentive system and less technical/financial risks, and to attract more external users from multiple blockchains outside Cosmos ecosystem

c. Inclusiveness

- The Cosmos Hub AMM should have strong philosophy of inclusiveness of users from different blockchains because its prime utility is inter-blockchain communication
- To possess such characteristics, the Liquidity module should provide most convenient ways for external users to come in and use the services provided by the Cosmos Hub
- The Liquidity module should not anticipate specific assets, such as Atom, into the process of user-flow in a forced manner. It is repeatedly proved that unnatural anticipation of native coin at unavoidable parts of process resulting in poor user attraction

d. Combination of Traditional Orderbook-based Model and New AMM Model

- Still new AMM model has multiple advantages over orderbook-based model, combination of both models will create more enriched utilities for wider potential users
- We re-define the concept of “swap order” in AMM as the “limit order with short lifetime” in an orderbook-based exchange. Then, two concepts from two different models can be combined as one united model so that the function can provide both ways to participate into the trading and liquidity providing activities
- Although our first version of the Liquidity module will not provide limit order option, but the base structure of the codebase is already anticipating such feature expansion in near future

e. Fair Trading Environment : Minimizing Low-Latency Competition

- In the traditional financial system, ordinary traders have significant disadvantages against institutional traders because of their inability to trade in a sub-second time frame. Lots of financial institutions around the world are making billions of profit from taking advantage of their experienced technology, better network equipment, and their business position advantage regarding information interception.
- This low-latency competition is revisited in blockchain communities which is revealed as “front-running”, “fake orders” and “manipulated transactions”
- We think that the competition of speed is a completely unnecessary characteristic of a DeX, so we adopted the “batch execution” model. Swap executions are processed every end of a batch, which can be of multiple number of blocks, and all messages inside the batch are handled fairly without any prioritized sorting by time dimension
- This approach will greatly reduce the risk of front-running problems, hence fairer trading environment

3. Liquidity Module Mechanism

a. Liquidity Pool Creation

- Permissionless : Anyone can create a liquidity pool
- Liquidity Pool Type : There can exist multiple types of liquidity pools so that the Liquidity module can provide various liquidity pool models in the future. The Liquidity module version 1 will only have the constant product type with two reserve coins
- Spam prevention : There exists a fee to create a liquidity pool to prevent spams

b. Liquidity Pool Deposit/Withdraw

- Deposit/Withdraw with Pool's Reserve Ratio : Deposits and withdrawals are accepted with pool's reserve ratio, just like Uniswap model
- Pool Token : Depositor receives pool tokens, which represent the right to withdraw a proportional amount of pool's reserve coins. Pool tokens are transferable

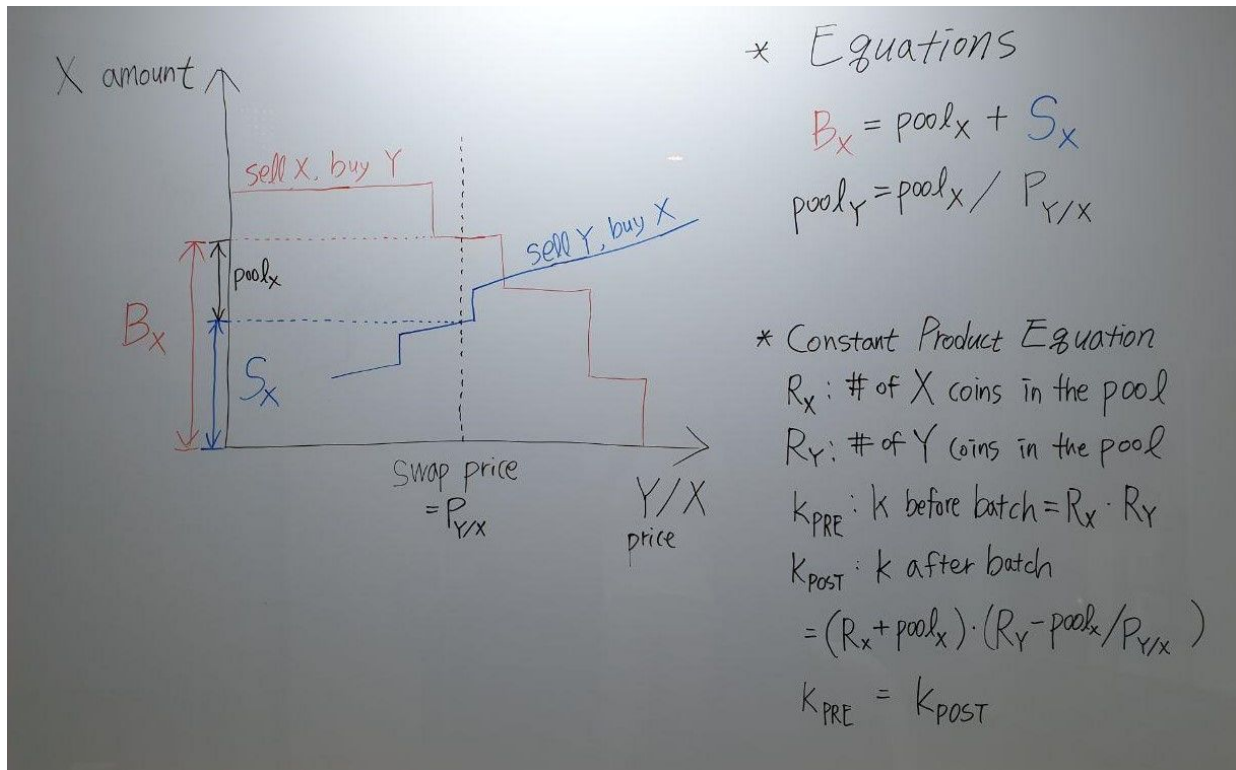
c. Swap Order

- Order Price : The swap order submitter sends the order price to the Liquidity module, which implies the maximum swap price where the submitter can accept for
- If the batch execution result cannot include the swap request with allowed range of swap price by the submitter, the swap order is automatically cancelled
- Compatibility with Orderbook-based Limit Order
 - The swap order interface is completely identical to limit order of orderbook-based model. This design will allow us to simply adopt orderbook-based limit orders without any design reconstruction (If the unexecuted swap orders are roll-overed to next batch, this is exactly what limit order is!)

d. Batch Concept

- Batch participants
 - Swap order submitters
 - The Liquidity Pool
- Conditions to be met
 - 1) All swap orders with acceptable swap price range are matched
 - 2) Constant product equation holds for the liquidity pool participation
- The swap price existence and uniqueness
 - Below calculation explains why the swap price with given conditions uniquely exists

e. Batch Calculation



1) View Swap Orders as Limit Orders

- Swap from X to Y : Limit order buying Y from X
- Swap from Y to X : Limit order selling Y for X

2) Order Amount Accumulation

- Priority sorting
 - Buy(Y coins) : Orders with higher Y/X prices have priority
 - Sell(Y coins) : Orders with lower Y/X prices have priority
- Accumulation Sum
 - B_X : Sum of X amount of buy orders accumulated from high to low price
 - S_Y : Sum of Y amount of sell orders accumulated from low to high price

3) Finding Swap Price

- Objective : To find the swap price which satisfies below constraints
 - constraint 1 : $B_X = \text{Pool}_X + S_X = \text{Pool}_X + S_Y \cdot P$
 - constraint 2 : $R_X \cdot R_Y = (R_X + \text{Pool}_X) \cdot (R_Y - \text{Pool}_X / P)$
- Uniqueness of the Solution
 - From constraint 1
 - $\text{Pool}_X = B_X(\text{non-increase}) - S_Y(\text{non-decrease}) \cdot P(\text{increase})$
 - Pool_X is a continuously decreasing variable over P
 - From constraint 2
 - $\text{Pool}_X = P(\text{increase}) \cdot R_Y(\text{fixed}) - R_X(\text{fixed})$
 - Pool_X is a continuously increasing variable over P
 - There exists unique P where both constraint 1&2 holds

4) Simulate Batch Execution

- With found swap price P, it is very easy to simulate the actual execution under the two given constraints 1&2