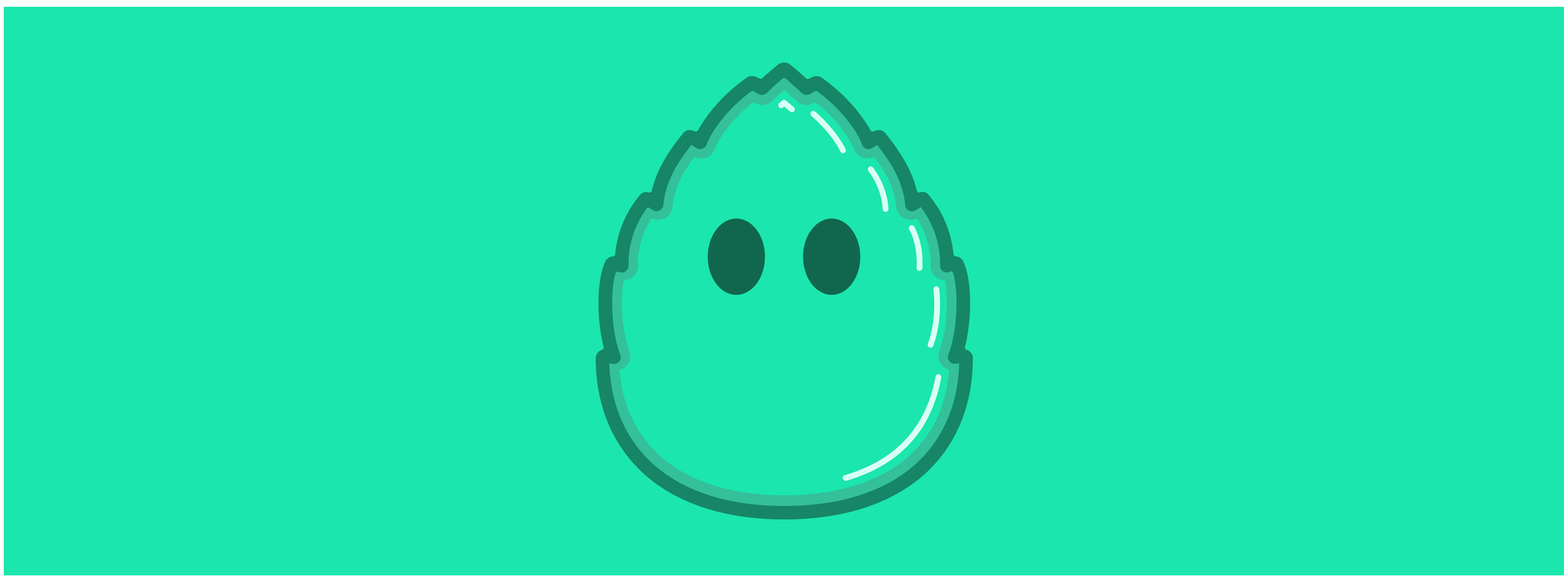
MintySwap version 1.0

July 1, 2021



Abstract

In this paper we present MintySwap - the constant function market maker with the most advanced protocol in the industry. It combines most of the well-recognized AMM industry standards together with more beneficial community-oriented features. In particular, we believe that the innovative system of fees will be more efficient for all the market participants.

1 Introduction

Traditional centralized stock exchanges typically have third party brokerage, unsecure environment and high transaction fees. TheBlockchain helps to create a more efficient trading environment for every market agent by creating a peer-to-peer trading environment that does not require a central exchange. Many Ethereum-based systems have appearedlately. Their usage can be obviously justified by the removal of third parties from the picture, reducing the time taken to settle orders and bringing down the transaction costs (see, e.g. [SMS20]).

One of the most popular ways to provide enough organic liquidity to support active trades on such exchanges is to use automatic market makers (AMM). In AMM a conservation function is used which helps to determine asset price based on given algorithm: it allows the exchange rates to change according to predetermined paths depending on the available assets quantities. This concept has been extensively studied in academic literature ([OS11], [Oth12]) the majority of which were primarily designed for information aggregation and implemented in markets where payoffs depend on some future state of the world (e.g. prediction markets). The most popular AMM are Logarithmic Market Scoring Rule, Bayesian market makers and dynamic pari-mutuel market makers ([Han03]). Decentralized exchanges for digital assets typically use Constant Function Market Makers. The term "constant function" refers to the fact that any trade must change the reserves in such a way that the product of those reserves remains unchanged.

This whitepaper covers the features of MintySwap - decentralized exchange that enables market paricipants to trade tokens on the Ethereum Blockchain. It does not require assets to be deposited to other smart contracts for trading to happen. Based on constant function market maker model it is able to constantly provide liquidity, no matter how big is the order size nor how small is the liquidity pool. Liquidity providers get rewards from the trading fees (paid by traders) proportional to the percentage their funds make of the entire pool. The design proposed in this paper uses the most recent findings in AMM industry together with the innovative system of fees for the DEX users.

2 Protocol Design

2.1 AMM

MintySwap uses Constant Function Market Makers (CFMM) class of AMM. It is based on a function which sets pre-defined prices using known quantities of given tokens. Compared to order book-based exchanges, agents trade against a pool of assets. Such pools fix the problem of limited liquidity. AMM offeres liquidity providers the incentive to supply these pools with tokens. DEX users get instant liquidity without having to seek for an exchange contact first, at the same time the providers of liquidity benefit from asset supply with exchange fees from pool users.

As a result of the work of arbitrageurs such model accurately reflects the price of tokens on reference markets. Namely, in case of any discrepancies, arbitrageurs will buy the asset on the CFMM and sell it on an order book-based exchange for a profit.

The protocol used in MintySwap allows to exchange tokens using one liquidity pool that contains both assets. To determine parameters of a product function the number of the product of both asset quantities is defined for all future states of the world. It happens at the launch of the particular pool. Every trade should preserve constant product after updating asset amounts in the particular pool.

In general, the AMM functionality can be formalized by a set of several mechanisms which help to answer how users can interact with the protocol and what the response of the protocol will be given actions of a particular user. Such mechanisms define how assets swapping and liquidity providing/ withdrawing are implemented. At the same time protocol-related properties help to determine how to calculate fees and rewards ([XVPC21]).

Let R_j denote the quantity of token j in the pool, P_j the current spot price of token j, $\mathcal{F}$ the conservation function invariant(s). Our object of interest is the state of liquidity pool denoted as

$$\omega = (\{R_j\}, \{P_j\}, \mathcal{F}).$$

The general rules of DEX based on AMM are the following: 1) The assets prices in the pool stay constant for pure liquidity provision and withdrawal activities; 2) The invariant of an AMM pool, $\mathcal{F}$, stays constant for pure swapping activities. The state transition after pure liquidity change can be expressed in the form:

$$(\{R_j\}, \{P_j\}, \mathcal{F}) \to (\{R_j^*\}, \{P_j\}, \mathcal{F}^{\dashv \int \sqcup})$$

and after asset swap:

$$(\{R_j\}, \{P_j\}, \mathcal{F}) \to (\{R_j^*\}, \{P_j^*\}, \mathcal{F}).$$

Note that the asset spot price can stay the same only if assets are summed up to or removed from a pool proportionate to the current reserve ratio $(R_1 : R_2 : ... : R_n)$.

Bonding curve can be expressed explicitly as a relational function between AMM invariant and reserve quantities

$$\{R_j\}, j = 1, ..., n, \mathcal{F} = F(\{R_j\}).$$

A conservation function for each token pair must be nonnegative, nondecreasing and concave.

The spot exchange rate between tokens t_j and token t_0 can be calculated as the slope of the bonding curve using partial derivatives of the conservation function F:

$$E_{0,i}(\{R_j\}, F) = \frac{\partial F/\partial R_0}{\partial F/\partial R_i}, \quad E_{0,0} = 1.$$

The amount Δ_0 of token t_0 received given amount Δ_i of token t_i can be calculated in two steps: 1) Adding input quantity Δ_i to the existing reserve of token t_i while the reserve quantity of any token other than token t_i or token t_0 stays the same; 2) using previous step the new reserve quantity of all tokens except for token t_0 can be computed. We can find the unknown new quantity R_0^* by plugging it in the conservation function:

$$F(\{R_j^*\}) = c.$$

One more characteristics slippage S_i measures the deviation between effective exchange rate Δ_i/Δ_0 and the pre-swap spot exchange rate $E_{0,i}$, namely ;

$$S_i = \frac{\Delta_i/\Delta_0}{E_{0,1}} - 1.$$

Further, the divergence loss L describes the loss in value of the all reserves in the pool compared to holding the reserves outside of the pool, after an asset price changes. Denote the increase of token t_0 by δ. Based on the formulas for spot price and swap quantity, L, can generally be computed using formula:

$$L = L(\delta, \{R_j\}) = \frac{W^*(\delta, \{R_j\})}{W_{out}(\delta, \{R_j\})} - 1.$$

Here $W^*(\delta, \{R_j\})$ is the new value of the pool computed as

$$\sum_k E_{i,k}(\{R_j^*\}) \cdot R_j^*.$$

By $W_{out}(\delta, \{R_j\})$ we denote the asset reserve value if it is outside of the pool:

$$W_{out}(\delta, \{R_j\}) = W(\{R_j\}) + E_{0,i}(\{R_j\}, F) \cdot R_0 \cdot \delta.$$

Note that DEX users and arbitrageurs constantly re-balance the pool through trading. Thus, asset value changes are reflected in exchange rate changes implied by the dynamic pool composition. As a corollary, the exchange rate between token t_0 and each token t_j implied by new reserve quantities R_j^* satisfy the following equations:

$$\delta = \frac{E_{0,1}(\{R_j^*\})}{E_{0,1}(\{R_j\})} - 1, \quad j \neq 0$$
$$F(\{R_j^*\}) = c$$

On MintySwap users supply liquidity pools with tokens and the price of the tokens in the pool is determined by a mathematical formula:

$$(R_1 - \Delta_1)(R_2 + f\Delta_2) = c.$$

Here R_1 and R_2 are reserves of each asset and f is the transaction fee. Trading any amount of either asset must change the reserves in such a way that, when the fee is zero, the product $R_1 \cdot R_2$ remains equal to the constant c.

Given the above-mentioned formula encoded in the smart contract of the pool, the implied assets spot price can be found based on the ratio between their reserve quantities. Namely, the spot exchange rate is simply

$$E_{1,2} = \frac{R_1}{R_2}.$$

The quantity of swapped tokens t_0 is the difference between the old and new reserve accounts:

$$\Delta_1 = R_1 - R_1^*$$

The slippage that a particular user will experience equals

$$S_1 = \frac{\Delta_1}{R_1}.$$

The reserve value of token t_1 equals

$$W_1 = \frac{W}{2} = W_2 = R_1.$$

When a liquidity provider have held R_1 tokens t_1 and R_2 tokens t_2, then when token amount t_2 increases by δ , the total value W_{out} is equal to

$$W_{out} = W + W_2 \cdot \delta = R_1 \cdot (2 + \delta).$$

With $R1$ of t_1 and R_2 of t_2 locked in a liquidity pool from moment of its creation, their quantity ratio would have been updated due to users' swapping to result in t_2's price change of δ. The updated pool value W^* will be

$$\frac{W^*}{2} = W_1* = R_1 \cdot \sqrt{1 + \delta}$$

Thus, the divergence loss can be expressed as a function of price change

$$L(\delta) = \frac{\sqrt{1 + \delta}}{1 + \delta/2} - 1$$

2.2 Fees

Anyone in possession of any type of ERC-20 tokens can become a liquidity provider by supplying tokens to an AMM's liquidity pool. Fee f paid by traders interacting with liquidity pool goes to liquidity providers proportional to the percentage their funds make of the entire pool.

Values R_1 and R_2 can be thought as virtual reserves of the pair of tokens in the pool. Thus, the formula for finding the amount of the token t_1 sent out after the trade is the following:

$$R_{1,final} = \frac{R_1 \cdot R_2}{R_2 + f\Delta_2}.$$

Instead of virtual reserves R_1 and R_2 the real contract tracks liquidity λ and price P. The formulas describing the realtionship are the following:

$$\frac{1}{\sqrt{P_i}} - \frac{1}{\sqrt{P_{i-1}}} = \frac{\Delta_1}{\lambda}$$

and

$$\Delta_2 = (\sqrt{P_i} - \sqrt{P_{i-1}}) \cdot \lambda.$$

Here index s represents tick. That is $(\sqrt{P_s} - \sqrt{P_{s-1}})$ is a change in $\sqrt{P}$.

MintSwap proposes an innovative system of fees allocation for DEX users. The default trading fee charged by MintSwap is $\varphi = 0.2\%$. This fraction of all trade volume is distributed proportionally to all liquidity providers. By default, these fees are put back into the liquidity pool, but can be collected by liquidity providers any time. What distinguishes Mintswap's approach to collecting fees from other DEXes is the ability to use token staking for maximizing the benefits from trading. Once you have staked your assets you can account for decrease in fees you have to pay each trade. Namely, the fee discount δ_j of agent j depends on two parameters: the time duration of staking Δt_j in years and the magnitude of stake n_j in shares of token emission. The relation is the following:

$$\delta_j = f(\Delta t_j, n_j) = \begin{cases} 1 \text{ if } n_j \geq 0.01, \Delta t_j \geq 1, \\ 0.5 \text{ if } n_j \geq 0.01, 1 > \Delta t_j \geq 0.5, \\ 0.25 \text{ if } n_j \geq 0.01, 0.5 > \Delta t_j \geq 0.25, \\ 0 \text{ if } n_j < 0.01 \text{ or } \Delta t_j < 0.25 \end{cases}$$

Thus, an individual trading fee f_j is computed using the formula:

$$f_j = (1 - \delta_j)\varphi.$$

3 Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. The opinions reflected herein are subject to change without being updated.

References

[Han03] Robin Hanson. Combinatorial information market design. *Information Systems Frontiers*, 5(1):107–119, 2003.

[OS11] Abraham Othman and Tuomas Sandholm. Automated market makers that enable new settings: Extending constant-utility cost functions. In *International Conference on Auctions, Market Mechanisms and Their Applications*, pages 19–30. Springer, 2011.

[Oth12] Abraham M Othman. Automated market making: Theory and practice. 2012.

[SMS20] Sashank Sridhar, Siddartha Mootha, and Sudha Subramanian. Decentralized stock exchange implementation using ethereum. In *2020 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 234–241. IEEE, 2020.

[XVPC21] Jiahua Xu, Nazariy Vavryk, Krzysztof Paruch, and Simon Cousaert. Sok: Decentralized exchanges (dex) with automated market maker (amm) protocols. *arXiv preprint arXiv:2103.12732*, 2021.