# The Gridex Protocol

November 2022 [Updated January 2023]

contact@gridex.org

### Abstract

The Gridex Protocol is a permissionless and non-custodial trading protocol consisting of a set of persistent, non-upgradable smart contracts on the Ethereum blockchain. Unlike the current mainstream decentralized exchanges based on the Automated Market Maker (AMM) model, Gridex is based on order books. With our novel Grid Maker Order Book (GMOB) model, we have significantly reduced the resource consumption of running an order book based system, allowing us to deploy and run it on the Ethereum blockchain while making the gas cost comparable to that of AMMs.

## 1. Introduction

Since 2020, the surge of decentralized exchanges (DEXs) based on the Automated Market Maker (AMM) model – or more precisely, the Constant Function Market Maker (CFMM) [1] exemplified by Uniswap; has led to the AMM model becoming the norm of DEXs. The minimal resource consumption of the AMM model compared to the Central Limit Order Book (CLOB) model − mainly utilized in traditional exchanges, makes it the key to easily implementing an on-chain system of smart contracts. DeFi would not be where it is without AMM, but it cannot meet the diverse needs of traders and market makers due to impermanent loss [2], high slippage, and lack of flexibility. Particularly in the AMM model, the greater the market volatility, the greater the risk of impermanent loss faced by liquidity providers (LPs). Consequently, the AMM model is most suitable for less volatile trading pairs, such as swaps between stablecoins [3].

The algorithms of matching engines, commonly implemented in centralized exchanges (CEXs) based on the CLOB model, consume a large number of resources making it difficult to deploy and run on the Ethereum blockchain. Some DEXs have attempted to solve this issue by implementing matching engines and order books off-chain [4] [5]. However, such solutions not only increase the risk of centralization (to a certain extent) but also limit their ability to interact with other protocols, making it impossible to utilize or contribute to the open ecosystem of Ethereum effectively. Henceforth, we have created the Grid Maker Order Book (GMOB) model and the Grid Price Linear Movement (GPLM) algorithm, which we describe in Sections 2 and 3, respectively. Compared with the algorithm of matching engines, GPLM is simplistic: resource consumption is significantly reduced, and it is easily implemented in smart contracts on the Ethereum blockchain.

## 2. Grid Maker Order Book

The Gridex Protocol is based on the GMOB model. In the GMOB model, maker orders are similar to limit orders in the CLOB model. However, there are three distinct differences:

1. Maker orders in the GMOB model are bounded within a very narrow price range (percentage based) instead of a specific price. The percentage of a range is called the 'resolution' (Section 4). The upper and lower boundaries of every range are called the 'boundaries'. Boundaries are similar to the ticks in Uniswap V3 [6] but have different usage.
2. Unlike limit orders, maker orders are never executed immediately and are added to the order book in the GMOB model. As maker orders add liquidity to the order book, the Gridex Protocol awards maker orders with negative trading fees. Therefore, traders will be granted a maker fee when a maker order is executed (Section 4).
3. A significant difference on the operational level between maker orders in the GMOB model and limit orders is that once the maker orders are filled, the user must manually execute the 'Collect' operation to get back the filled assets and the rewarded fees.

## 3. GPLM Algorithm

In this section, we explain how a taker order (swap) functions. We define the following terms as:

| Terminology | Notes |
|---|---|
| $\boldsymbol{token0}$ | – |
| $\boldsymbol{token1}$ | – |
| $\boldsymbol{zeroForOne}$ | A taker uses $token0$ to exchange for $token1$ |
| $\boldsymbol{oneForZero}$ | A taker uses $token1$ to exchange for $token0$ |
| $\boldsymbol{P}$ | The price of $token0$ in terms of $token1$ |
| $\boldsymbol{P_n}$ | The new price after a taker order has been filled |
| $\boldsymbol{P_c}$ | Current price |
| $\boldsymbol{P_a}$ | The average transaction price of a taker order |
| $\boldsymbol{P_b}$ | When the trading direction is $zeroForOne$, $P_b$ is the lower boundary of the range<br>When the trading direction is $oneForZero$, $P_b$ is the upper boundary of the range |
| $\boldsymbol{M}$ | When the trading direction is $zeroForOne$, $M$ is the amount of $token1$ from all maker orders in the current range<br>When the trading direction is $oneForZero$, $M$ is the amount of $token0$ from all maker orders in the current range |
| $\boldsymbol{T_o}$ | The amount of tokens received by the taker (outputted) |
| $\boldsymbol{T_i}$ | The amount of tokens submitted by the taker (inputted) |
| $\boldsymbol{exactInput}$ | Output calculated based on input by the taker |
| $\boldsymbol{exactOutput}$ | Input calculated based on output by the taker |

Table 1: Terminology Table

When $T_o$ is less than or equal to $M$, we use the following equation to calculate the price after the transaction:

$$P_n = P_c + \frac{T_o\,(P_b - P_c)}{M}$$

Eq. (3.1)

We use the following equation to calculate the average transaction price for the taker:

$$P_a = \frac{P_n + P_c}{2}$$

Eq. (3.2)

When the taker uses the $exactOuput$ operation, the following equations are used to calculate $T_i$:

$zeroForOne$

$$T_i = \frac{T_o}{P_a}$$

Eq. (3.3)

$oneForZero$

$$T_i = T_o\,P_a$$

Eq. (3.4)

When $P_n$ is specified by the taker, Equation 3.1 is used to calculate $T_o$:

$$T_o = \frac{M\,(P_n - P_c)}{P_b - P_c}$$

Eq. (3.5)

When the taker uses $exactInput$, and the direction of the trade is $zeroForOne$, by using Equations 3.1, 3.2 and 3.3 we have:

$$T_o = \frac{2MT_iP_c}{2M - T_i\,(P_b - P_c)}$$

Eq. (3.6)

When the taker uses $exactInput$, and the direction of the trade is $oneForZero$, Equations 3.1, 3.2 and 3.4 result in:

$$T_o = \frac{M\sqrt{P_c^2 + \frac{2T_i(P_b - P_c)}{M}} - MP_c}{P_b - P_c}$$

Eq. (3.7)

The square rooting operation consumes a large amount of resources on-chain. To save gas cost, we calculate $T_o$ in the $oneForZero$ case by redefining $P \rightarrow \frac{1}{P}$ in Equation 3.6. We obtain:

$$T_o = \frac{\frac{2MT_i}{P_c}}{2M - T_i\,(\frac{1}{P_b} - \frac{1}{P_c})}$$

Eq. (3.8)

'Linear' in the Grid Price Linear Movement (GPLM) algorithm is representative of the fact that within a price range, the price change caused by a taker order is linearly proportional to the quantity of tokens taken away.

A typical case of how the GPLM algorithm functions is illustrated in the figure below.

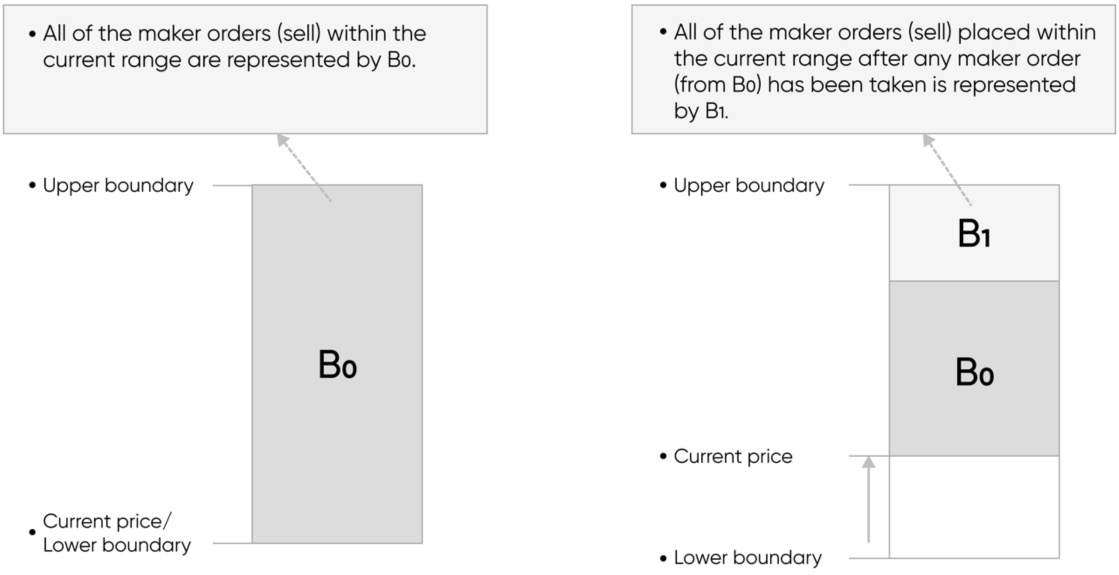
• All of the maker orders (sell) within the current range are represented by B0.
• Upper boundary
B0
• Current price/ Lower boundary
• All of the maker orders (sell) placed within the current range after any maker order (from B0) has been taken is represented by B1.
• Upper boundary
B1
B0
• Current price
• Lower boundary

Figure 1: A Typical Case of the GPLM Algorithm

As shown in the figure, when a user exchanges $token1$ for $token0$ ($oneForZero$), the price of $token0$ will rise after the exchange.

In the following steps, maker and taker orders are sell and buy orders, respectively.

1. We assume that the current price ($P_c$) is positioned precisely on the lower boundary of the price range. We express all the maker orders in this price range as bundle 0 ($B_0$).
2. A taker order is placed where the buy amount ($T_o$) is less than the total amount of maker orders in the current range ($M$). The price after the transaction is given by Equation 3.1. The amount of each maker order in $B_0$ will be filled by $\frac{T_o}{M}$. The average filled price of the taker order can be formulated by using Equation 3.1 in conjunction with Equation 3.2. If the buy amount ($T_o$) exceeds $M$, proceed directly to step 4.
3. After $B_0$ is first traded by a taker, if there are still users who place maker orders in this range, these new orders are expressed as bundle 1 ($B_1$). When all maker orders of $B_0$ are filled by takers, a check is performed to see whether $B_1$ exists. If so, $B_1$ is replaced by $B_0$, and steps 2 and 3 are repeated until there are no more maker orders in the current range; then, the algorithm proceeds to step 4. If not, directly proceed to step 4.

4. When all maker orders in the range are executed, and the taker order which bought out the last maker orders is not entirely filled, the price will touch the upper boundary of the range and enter the next range. Then, repeat from step 1 to calculate the future price movement.

## 4. Grids and Fee Levels

The Gridex Protocol will initially support 3 types of grids:

| Grid Resolution ($G\%$) | Maker Fee | Taker Fee |
|---|---|---|
| 0.01% | −0.01% | 0.01% |
| 0.05% | −0.05% | 0.05% |
| 0.3% | −0.3% | 0.3% |

Table 2: Grids and Fee Levels

A grid resolution ($G\%$) defines the step size ($100G$) used to create boundary values $P_b(i)$, which are indexed by integer values $i$. These can be determined by the following functions:

$$P_b(i) = 1.0001^{(100G)i} \tag{4.1}$$

Eq. (4.1)

Similarly, $i$ can be derived from a given $P_b$ and $G\%$:

$$i = \frac{1}{100G}\log_{1.0001}(P_b)$$

Eq. (4.2)

## References

[1] Guillermo Angeris and Tarun Chitra. 2020. Improved Price Oracles: Constant Function Market Makers. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (AFT' 20).* Association for Computing Machinery, New York, NY, United States, 80–91. https://doi.org/10.1145/3419614.3423251

[2] Stefan Loesch, Nate Hindman, Nicholas Welch and Mark. B. Richardson. 2021. *Impermanent loss in uniswap V3.* Retrieved Nov 01, 2022 from https://arxiv.org/pdf/2111.09192.pdf

[3] Michael Egorov. 2019. *StableSwap – efficient mechanism for Stablecoin liquidity.* Retrieved Nov 01, 2022 from https://classic.curve.fi/files/stableswap-paper.pdf

[4] Will Warren and Amir Bandeali. 2017. *0x: An open protocol for decentralized exchange on the Ethereum blockchain.* Retrieved Nov 01, 2022 from https://www.0x.org/pdfs/0x_white_paper.pdf

[5] Antonio Juliano. 2017. *dYdX: A Standard for Decentralized Margin Trading and Derivatives.* Retrieved Nov 01, 2022 from https://whitepaper.dydx.exchange/

[6] Hayden Adams, Noah Zinsmeister, Moody Salem, River Keefer, and Dan Robinson. 2021. *Uniswap v3 Core.* Retrieved Nov 01, 2022 from https://uniswap.org/whitepaper-v3.pdf

## Disclaimer

This paper is for general information purposes only. It does not constitute investment advice, a recommendation, or a solicitation to buy or sell any investment. It should not be used to evaluate the merits of making any investment decision. It should not be relied upon for accounting, legal, tax advice, or investment recommendations.