Whitepaper

DODO

———A Next-Generation On-Chain Liquidity Provider Powered by Pro-active Market Maker Algorithm

V1.0

DODO Team

contact@dodoex.io

Abstract: This article introduce DODO, a next-generation on-chain liquidity provider, which leverages the Proactive Market Maker algorithm (PMM) to provide pure on-chain and contract-fillable liquidity for everyone. Comparing to other on-chain liquidity solutions, DODO has multiple advantages: high fund utilization, low slippage, single risk exposure, reduced impermanent loss. We also discuss the core concepts and mathematical details about Proactive Market Maker algorithm (PMM), and include the contract framework.

Keywords: PMM; high fund utilization; reduced impermanent loss; single risk exposure

1 Introduction

DODO is a next-generation on-chain liquidity provider, which leverages the Proactive Market Maker algorithm (PMM) to provide pure on-chain and contract-fillable liquidity for everyone. DODO accepts liquidity providers' assets. It gathers funds near market prices to provide sufficient liquidity. In order to minimize counterparty risks for LPs, DODO dynamically adjusts market prices to encourage arbitrageurs to step in and stabilize LPs' portfolios. Comparing to other on-chain liquidity solutions, DODO has multiple advantages: high fund utilization, low slippage, single risk exposure, reduced impermanent loss.

As a trader, you can see these features: Each and every trader enjoys sufficient liquidity similar to that of centralized exchanges; Arbitrageurs can profit from price discrepancies between DODO and other exchanges; Smart contracts can natively use DODO liquidity to complete on-chain transactions, such as liquidation and auctions

As a liquidity provider (LP), you can see these features: There are no minimal deposit requirements and restrictions on asset types; DODO charges a fee for each transaction and eventually distributes it to LPs as rewards; LPs can create trading pairs with their own tokens; LPs can obtain liquidity by depositing their tokens they already own, without taking on price risk.

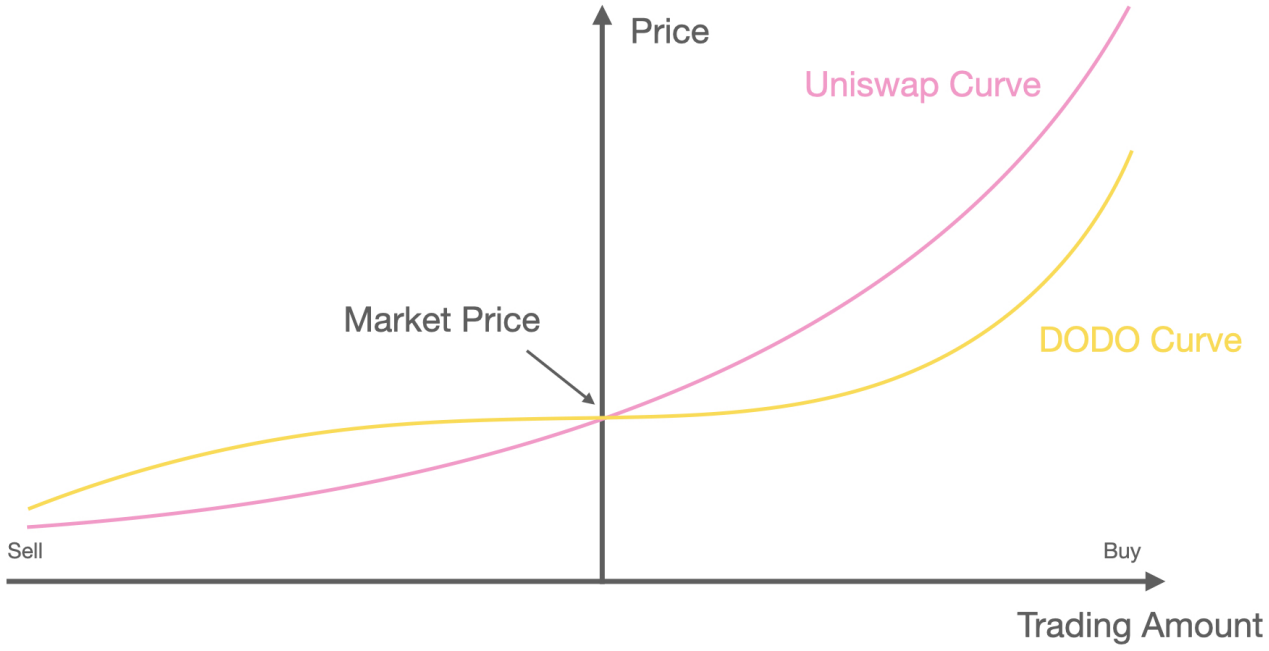
2 Protocol

2.1 The DODO Advantage

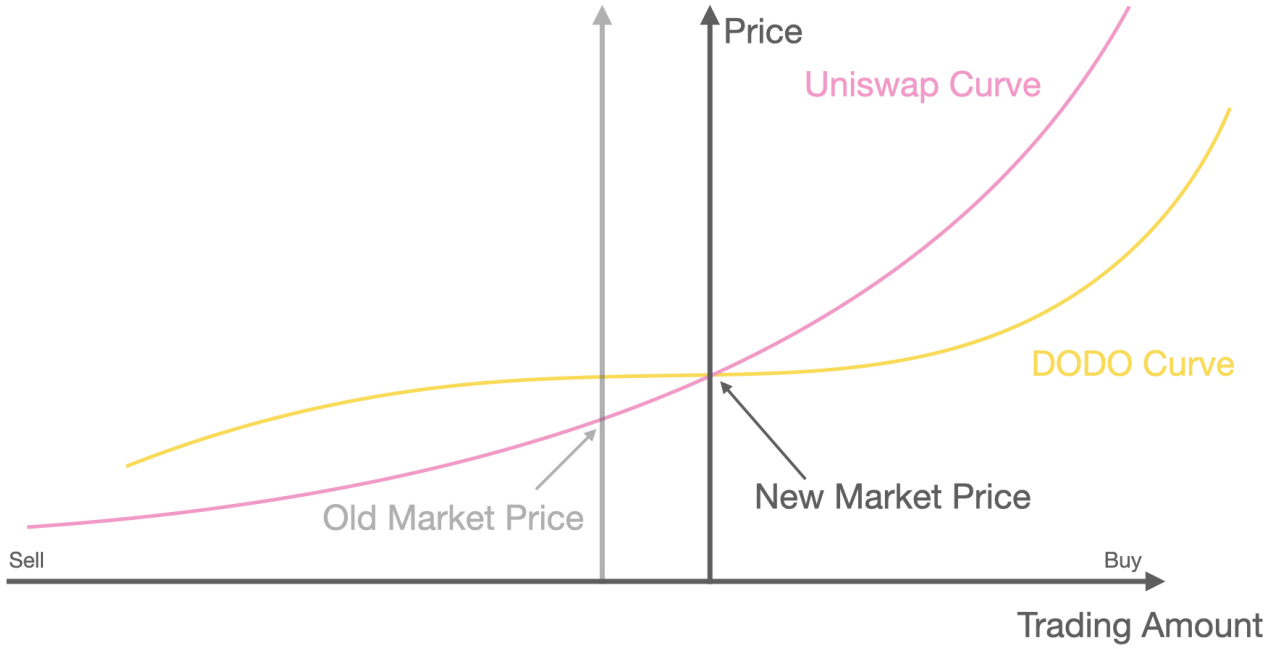
2.1.1 Overview

DODO is powered by a ground-breaking algorithm called Proactive Market Maker (PMM). PMM leverages price oracles to retrieve accurate market prices of assets as input. It then aims to provide sufficient liquidity near the market price for every asset. The result is that liquidity decreases rapidly when far away from the market price. The following graphs compare the price curves of DODO (PMM) and Uniswap (AMM).

With everything else fixed, it is clear that the PMM curve is significantly flatter than the AMM curve near the market price, indicating higher fund utilization and lower slippage. Prices provided by PMM are more favorable than AMM.



As the market price changes, AMM passively relies on arbitrage trading to change prices. On the other hand, PMM proactively shifts the price curve in the same direction to ensure that the section in the vicinity of the market price remains flat. This ensures the constant provision of sufficient liquidity.



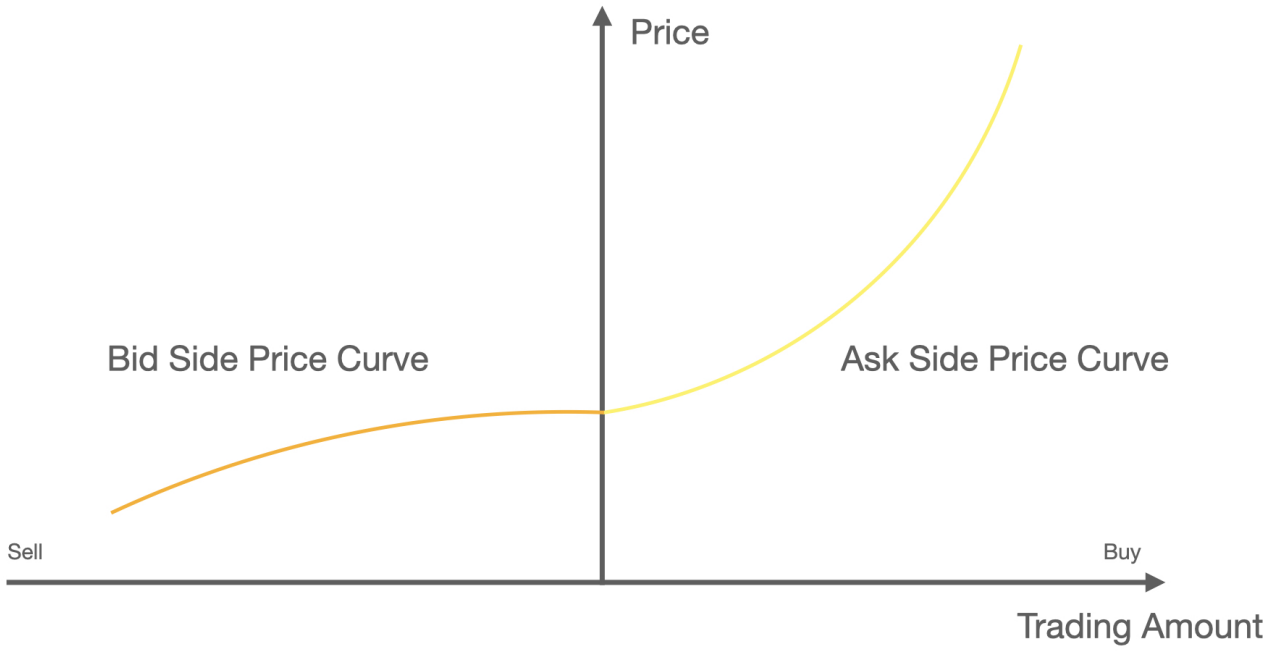
PMM outperforms AMM solutions in several important aspects.

- ***High Fund Utilization***

As seen in the above graphs, PMM, like AMMs, provides liquidity in the price range of zero to positive infinity, but the PMM price curve is significantly flatter in the area near the oracle (market) price. That is, most of the funds are gathered near the market price, which allows for more active, frequent trading, increasing fund utilization.

- ***Single Risk Exposure***

In the price curves above, each price curve consists of two parts: the bidding side to the left and the asking side to the right. These two parts may have different depth, or liquidity, resulting in what is known as the bid-ask spread[1].



In PMM, the asking liquidity is solely determined by the amount of base token in the pool, and the bidding liquidity is solely determined by the amount of quote tokens in the pool. It allows the base and quote pools tohave different sizes, and thus allows liquidity providers to deposit any amount of either quote or base tokens, rather than both (like Uniswap). DODO Liquidity providers deposit what they already have, nothing more.

Please note: The design is intuitive, because when you take an asking order, you take liquidity providers' base tokens and the quote tokens are irrelevant.

- ***Reduced Impermanent Loss***

But what about impermanent loss, i.e. how does PMM ensure that liquidity providers get what they deposited when they withdraw their tokens? The answer is by encouraging arbitrage trading. When individual traders buy base tokens, PMM slightly increases the price to make it more profitable for arbitrageurs to sell base tokens. In PMM, arbitrage trading makes sure that the number of tokens in the pool is always roughly equal to the number of tokens deposited by liquidity providers. This scheme effectively mitigates impermanent loss for liquidity providers, making liquidity provision on DODO a low-risk affair.

2.1.2 Next Generation of Liquidity Provision

Liquidity is the most important resource in the DeFi world, because it is the foundational element in all DeFi projects. There are two major proven approaches to decentralized liquidity provision today:

- Algorithmic market makers (e.g. Uniswap)
- Orderbook-based order matching (e.g. dYdX)

However, they are both flawed.

- Compared to centralized exchanges, algorithmic market makers cannot provide sufficient liquidity for mainstream assets. In addition, for niche, long-tail assets, AMM can only provide very basic liquidity support
- Orderbook-based order matching relies on human market makers to mirroring centralized exchanges liquidity. Effective market makers are expensive, and very few DEX teams can afford them. In addition, this kind of liquidity is difficult to be filled by smart contracts due to the human elements involved, significantly limiting the number of use cases for DeFi practitioners

PMM is also an algorithmic market maker algorithm, but it fundamentally differs from other approaches by mitigating and eliminating their disadvantages and amplifying their advantages. PMM provides sufficient and contract-fillable liquidity on-chain for all assets, empowering DeFi users to take advantage of composability

2 .2 Core Concepts

2.2.1 Base & Quote Token

Base _and _quote are two concepts that will be mentioned frequently. Two easy ways to distinguish between them are:

- In a trading pair, the *base* is always the token before the hyphen, and *quote* after
- In transactions, price refers to how many *quote* tokens are needed in exchange for one *base* token

For example, in the ETH-USDC trading pair, ETH is the *base* token and USDC is the *quote* token

2.2.2 PMM Parameters

The funding pool of PMM is described by four parameters:

- B_0: base token regression target - total number of base tokens deposited by liquidity providers
- Q_0 : quote token regression target - total number of quote tokens deposited by liquidity providers
- B: base token balance - number of base tokens currently in the pool
- Q: quote token balance - number of quote tokens currently in the pool

2.2.3 PMM Pricing Formula

The PMM price curve is plotted by the following pricing formula:

$$P_{margin} = iR$$

Where *R* is defined to be the piecewise function below:

if $B < B_0, R = 1 - k + (\frac{B_0}{B})^2 k$

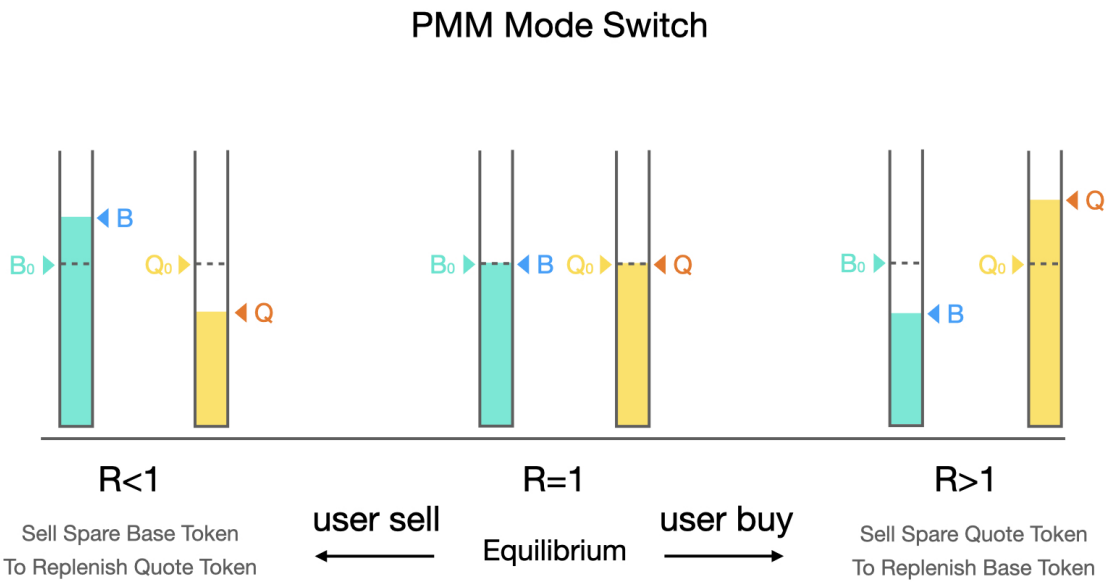
if $Q < Q_0, R = 1/(1 - k + (\frac{Q_0}{Q})^2 k)$

else $R = 1$

i is the market price provided by an oracle, and *k* is a parameter in the range (0, 1).

2.2.4 Three Possible States in PMM

At any given time, PMM is in one of three possible states: equilibrium, base token shortage, and quote token shortage.



Initially, i.e. prior to any transaction, the capital pool is in equilibrium, and both base tokens and quote token are at their regression targets. That is, $B = B_0$ and $Q = Q_0$.

When a trader sells base tokens, the base token balance of the capital pool is higher than the base token regression target; conversely, the quote token balance is now lower than the quote token regression target. In this state, PMM will try to sell the excess base tokens, lowering the base token balance and increasing the quote token balance, in order to move this state back to the state of equilibrium.

When a trader buys base tokens, the quote token balance of the capital pool is higher than the quote token regression target; conversely, the base token balance is now lower than the base token regression target. In this state, PMM will try to sell the excess quote tokens, lowering the quote token balance and increasing the base token balance, in order to move this state back to the state of equilibrium.

The parameter *R* in the pricing formula above assumes a critical role in facilitating this regression process. The more the capital pool deviates from the equilibrium state, the more *R* deviates from *1*. When the price given by the PMM algorithm deviates from the market price, arbitrageurs step in to help bring the capital pool back to the equilibrium state.

2.2.5 Liquidity Provider Fee

A small amount of transaction fee will be charged on every trade. This fee is called the liquidity provider fee and will be distributed to every liquidity provider based on their proportional stake in the capital pool.

More specifically, liquidity provider fees are collected from what buyers received and distributed to liquidity providers who supplied this kind of asset to the capital pool. In other words, liquidity providers are rewarded in the same asset denomination.

For example, when traders buy ETH tokens with USDC tokens, liquidity provider fees will be charged in the form of ETH tokens, and distributed to liquidity providers who deposited ETH tokens into the capital pool.

When traders sell ETH tokens for USDC tokens, liquidity provider fees will be charged in the form of USDC tokens, and distributed to liquidity providers who deposited USDC tokens into the capital pool.

Please note: Base and quote tokens have different returns on investments (ROI) in PMM's funding pool.

2.2.6 Maintainer Fee

A maintainer fee is also collected from what buyers received, and will be directly transferred to the maintainer. The maintainer may be a development team, a foundation, or a staking decentralized autonomous organization (DAO). Currently, the maintenance fee on DODO is 0.

2.2.7 Withdraw Fee

A withdrawal will change the PMM price curve and may harm the interests of other liquidity providers. DODO charges a withdrawal fee from liquidity providers who withdraw their assets and distribute it to all remaining liquidity providers.

Please note:

Normally, the withdrawal fee is 0 or an extremely small percentage (\<0.01%) of what you withdraw. The withdrawal fee will increase significantly only if the funding pool suffers from a serious shortage of either base or quote tokens and liquidity providers intend to withdraw the type of token in shortage. The withdrawal fee serves as a protection mechanism for liquidity providers who maintain their supplies of liquidity and contribute to the sustainability and overall health of the DODO platform.

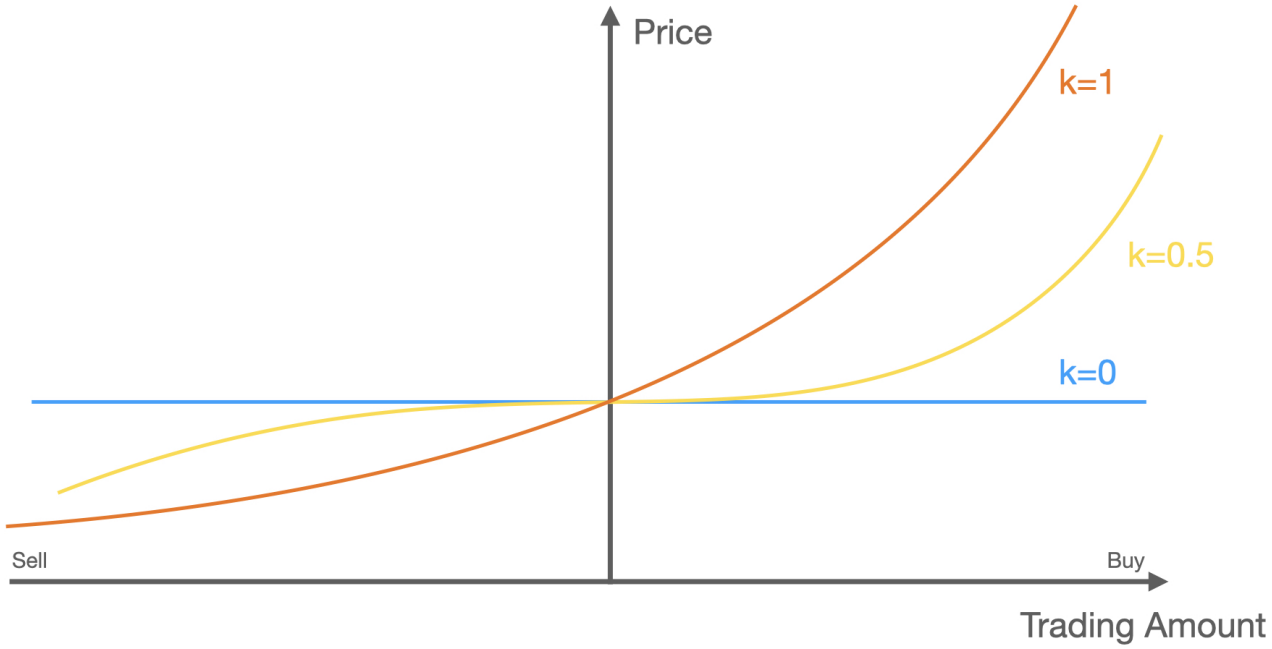
2.2.8 Deposit Rewards

Rewards will be distributed to those who make a deposit of base(quote) tokens when the capital pool faces a shortage of base(quote) tokens.

In the next section, we will explain the math behind these core concepts.

2.2.9 Flexibility and *k*, the "Liquidity Parameter"

Last but not least, we will introduce the DODO's "liquidity parameter", *k*. The parameter *k* gives DODO the flexibility to handle different market situations.



When *k* is *0*, DODO naively sells or buys at market price, as shown by the flat, blue line. As *k* increases, DODO's price curve becomes more "curved", but, consequently, liquidity becomes increasingly jeopardized, because more funds are placed far away from market price and are thus underutilized or not utilized at all. When *k* increases to *1*, the flat section near the market price is completely eliminated and the curve essentially becomes a standard AMM curve, which Uniswap uses.

Normally, *k* is recommended to be a relatively small value, such as *0.1*, which could provide liquidity 10 times better than the standard AMM algorithm.

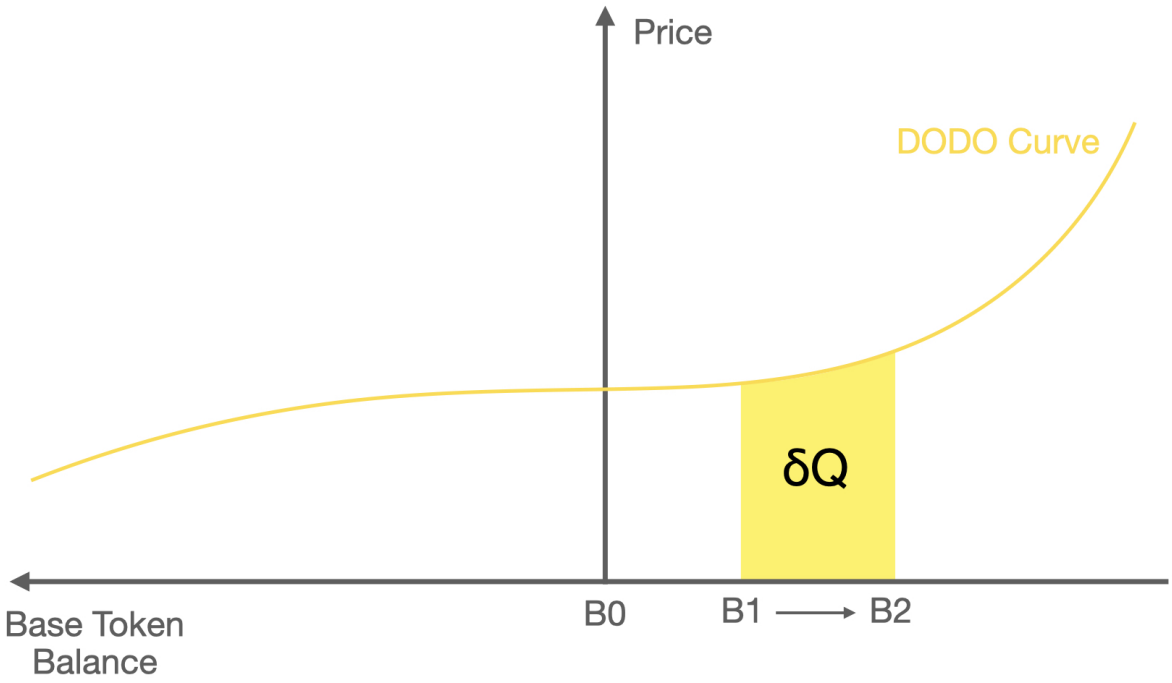
2.3 The Math Behind PMM

2.3.1 Core PMM

The core of PMM is essentially calculating one integral and solving two quadratic equations. The smart contract implementation can be found here[2].

2.3.1.1 The Price Curve Integral

For traders, the most important thing is the average transaction price. The average transaction price is the integral of the marginal price . Let's take the base token shortage scenario as an example.



$$\Delta Q = \int_{B_1}^{B_2} P_{margin} dB$$

$$= \int_{B_1}^{B_2} (1-k)i + i(B_0/B)^2 k dB$$

$$= i(B_2 - B_1) * (1 - k + k\frac{B_0^2}{B_1 B_2})$$

This tells the trader how much they should pay if they buy $B_1 - B_2$ amount of base tokens.

Rearranging the equation above, the average transaction price is thus:

$$P = \frac{\Delta Q}{B_2 - B_1} = i * (1 - k + k\frac{B_0^2}{B_1 B_2})$$

We found that the average transaction price is only dependent on the state of the system before and after the transaction, so the price calculation methods for both buying and selling are the same: integrating P_{margin} .

2.3.1.2 Solving the Quadratic Equation for Trading

Without the loss of generality, the integral becomes the following when there is a shortage of quote tokens:

$$\Delta B = \frac{1}{i}(Q_2 - Q_1) * (1 - k + k\frac{Q_0^2}{Q_1 Q_2})$$

Let's derive how to calculate the price when there is a shortage of quote tokens and only the number of base tokens you want to buy or sell (i.e. ΔB) is given.

Now that ΔB , Q_0 , Q_1 are given, we need to calculate Q_2 , which is found by solving a quadratic equation. Transforming the equation into standard form:

$$(1-k)Q_2^2 + (\frac{kQ_0^2}{Q_1} - Q_1 + kQ_1 - i\Delta B)Q_2 - kQ_0^2 = 0$$

let $a = 1 - k, b = \frac{kQ_0^2}{Q_1} - Q_1 + kQ_1 - i\Delta B, c = -kQ_0^2$

Because $Q_2 \geq 0$, we discard the negative root, and so $Q_2 = \frac{-b+\sqrt{b^2-4ac}}{2a}$

It can be proven that:

- When $\Delta B > 0$, $Q_2 > Q_1$; trader buy base token, and should pay $Q_2 > Q_1$
- When $\Delta B < 0$, $Q_2 < Q_1$; trader sell base token, and will receive $Q_2 > Q_1$
- When $\Delta B = 0, \quad Q_2 = Q_1$

2.3.1.3 Solving the Quadratic Equation for Regression Targets

When the system is not in the equilibrium state, changes to the oracle price will bring profit or loss. For example, assume that shortage of base tokens is the current state, and the oracle price goes up. It is clear that the excess quote tokens cannot buy enough base tokens to return the base token balance to the base token regression target. Thus, LPs who deposited base tokens will suffer a loss. Conversely, if the oracle price drops, the excess quote tokens can buy more base tokens, causing the base token balance to exceed the base token regression target, and LPs who deposited base tokens will make a profit.

In summary, the regression target is influenced by the oracle price. To calculate the regression target at a certain oracle price, we make the following derivation:

Given:

$$\Delta Q = i(B_2 - B_1) * (1 - k + k\frac{B_0^2}{B_1 B_2})$$

Since we are doing regression, . Rearranging the equation with respect to gives

$$\frac{k}{B_1}B_0^2 + (1-2k)B_0 - [(1-k)B_1 + \frac{\Delta Q}{i}] = 0$$

The negative root does not make sense and is discarded, so B_0 is:

$$B_0 = B_1 + B_1 \frac{\sqrt{1+\frac{4k\Delta Q}{B_1 i}}-1}{2k}$$

In this case, $\Delta Q = Q - Q_0$. It can be proven that, when $\Delta Q \geq 0$, $B_0 < B_1$.

This fact is extremely important, because it ensures that the base token balance and the quote token balance will never be greater than the regression target simultaneously, or less than the regression target simultaneously. This means that PMM will only switch between the three states discussed in the Core Concepts section.

Similarly, the formula for quote token regression target Q_0 is:

$$Q_0 = Q1 + Q1 * \frac{\sqrt{1+\frac{4k\Delta Bi}{Q_1}}-1}{2k}$$

2.3.2 Peripheral

This section will deal with the math pertaining to the peripheral functioning of PMM.

2.3.2.1 Trades

As mentioned above, the regression target depends on the oracle price, and the price curve in turn depend on the regression target. So in every trade, we should calculate the regression target well in advance to make the price curve fixed.

In addition, since the price curve given by PMM is segmented, if a transaction involves different states (for example, when a trader sells an astronomical amount of base tokens during a base token shortage and forces the state into a quote token shortage), the price needs to be calculated in segments as well.

Please be advised that this calculation requires a high degree of accuracy. The smart contract provides six trading functions for the three possible states. You can find the most important logic of cross-state trading here[3].

2.3.2.2 Deposit

Depositing and withdrawing base token when there is a shortage of base tokens, or quote tokens when there is a shortage of quote token, will change the price curve. This requires us to process the deposit and withdrawal with caution and care in order to keep the capital pool sustainable and fair.

We will analyze what happens when an LP makes a deposit when there is a shortage of base tokens.

According to the calculation formula of B_0 derived above

$$B_0 = B_1 + B_1 * \frac{\sqrt{1+\frac{4k\Delta Q}{{B_1}^2}}-1}{2k}$$

After an LP deposit *b* base tokens, B_1 increases by *b*, and B_0 increases more than *b*'s magnitude. It means that this deposit helps all LPs who provided base token make a profit. The reason why is that the deposit makes the price curve smoother, and the same amount of ΔQ can now buy more base tokens.

In this case, as soon as the LP makes a deposit, the LP makes a profit. This is referred to as the deposit reward. The essential source of this reward is the slippage paid by the trader who made the system deviate from equilibrium state.

Please note: It is important to note that deposit rewards are not risk-free arbitrage trading opportunities.

2.3.2.3 Withdrawal

Similarly, after an LP withdraws *b* base tokens, B_1 decreases by *b*, and B_0 decreases by more than b's magnitude. This withdrawal causes all LPs who owes Base Tokens to suffer losses. This is because this withdrawal makes the price curve more steep, and the excess quote tokens have less purchasing power in terms of base tokens.

The PMM algorithm stipulates that a withdrawal fee is required to withdraw tokens in this case. The magnitude of the fee is equal to the aggregate loss of all LPs caused by the withdrawal. This fee will be directly distributed to all LPs that have not yet withdrawn.

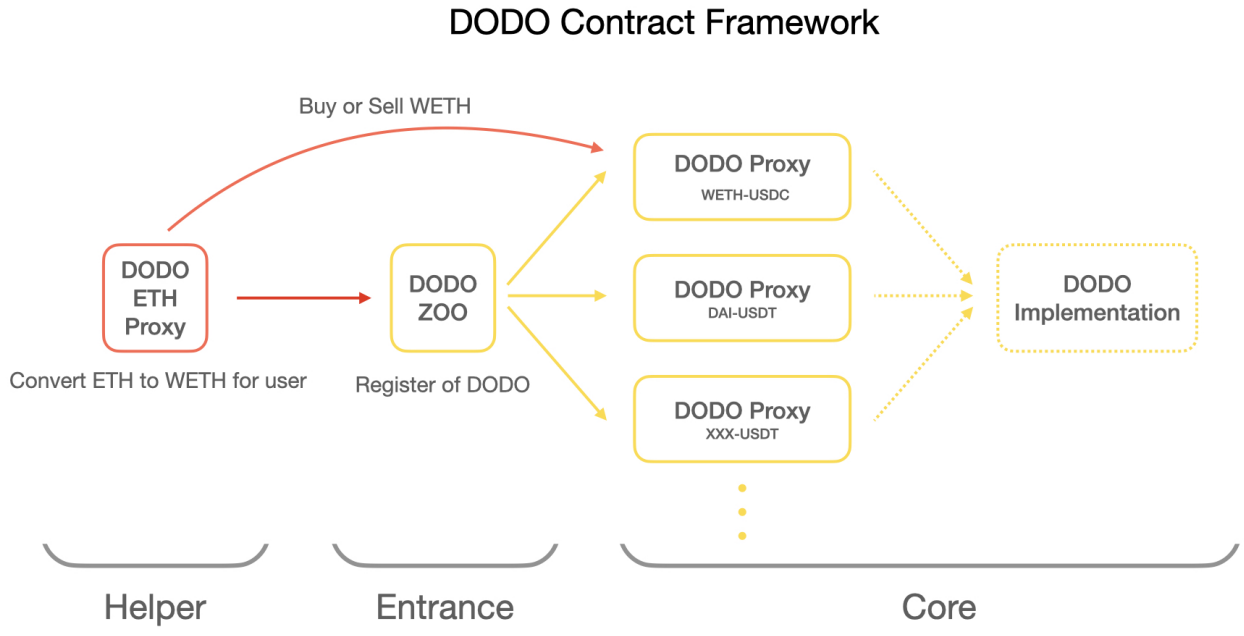
Factoring in the deposit reward from the previous section, if an LP makes a withdrawal immediately after depositing, the withdrawal fee will be greater than the deposit reward, thus eliminating any possibility of risk-free arbitrage trading.

It is worth noting that both deposit reward and withdrawal fee are only significant when the system deviates very far from the equilibrium state and the deposit/withdrawal amount is large. Traders thus often overlook the existence of this gain/loss. Of course, traders are also welcome to extract value from the system by exploiting this if they so wish. In order to do that, they can first deposit to earn deposit rewards when the system deviates from the equilibrium, and then withdraw once the system returns to the equilibrium to avoid the withdrawal fee.

3 Contract

3.1 Overview

DODO is built with a set of smart contracts. The following figure shows the framework of these contracts and how they interact with each other in the DODO architecture.



3.2 Core

The core part of the DODO framework, which contains all the data and logic of DODO, consists of a set of DODO Proxy contracts and a singular DODO Implementation contract. Each trading pair binds with an independent *DODO Proxy* contract (e.g. WETH-USDC, DAI-USDT, etc.), which is a transparent proxy that only stores states and metadata. All underlying logic lies in the *DODO Implementation* contract.

For convenience's sake, we will call the transparent proxy *DODO Pair* and the logic implementation *DODO Template*. Users should interact with *DODO Pair* directly or through *Helper*.

3.3 Entrance

DODO is an open-source contract, and the DODO team welcomes forks. However, it is important to note that the operation of *DODO Pair* is highly dependent on oracles and parameter fine-tuning, and a misconfigured *DODO Pair* could potentially cause significant losses for users. Therefore, we deployed an entrance contract to help blockchain developers navigate these obstacles. All *DODO Pairs* registered in this contract have been rigorously tested and audited, as the DODO team believes the safety of DODO users is of utmost importance. Developers should only look for the entrance called *DODO Zoo* when developing upon DODO. Even if the *DODO Template* is upgraded, *DODO Zoo* will remain unchanged.

3.4 Helper

There are a lot of tedious tasks that can be packaged using contracts to make them easy to use and understand. For example, the *DODO ETH Proxy* shown in the figure above helps users convert between ETH and WETH and interact with *DODO Pair*. This way, the underlying complexity with WETH is abstracted away from users, effectively protecting them - users do and should only care about directly buying or selling ETH on DODO. There are many such contracts, such as arbitrage and route, which we collectively call *Helper*. We invite the community to help develop more helper contracts and we are willing to provide guidance and support.

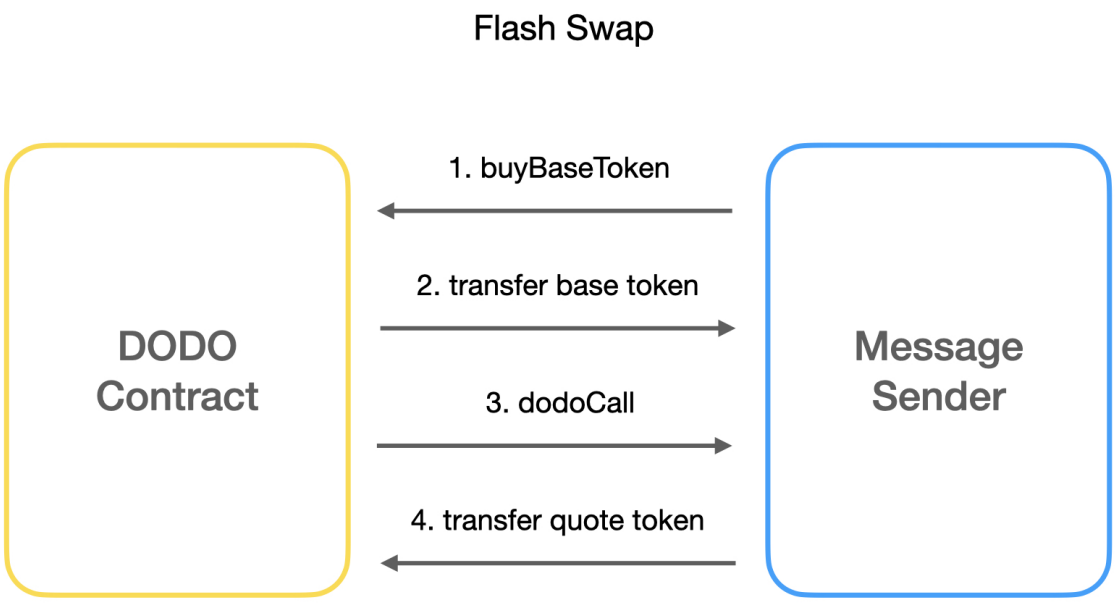
4 Use Case

4.1 Flash Swap

4.1.1 What is Flash Swap

Simply put, you are allowed to pay on credit on DODO! When you buy tokens DODO, you can first get the tokens you want to buy, do anything you want with the tokens, and pay for them later.

4.1.2 How Does Flash Swap Work



The figure above illustrates the four steps in a flash swap happening under the hood

1. Call the *buyBaseToken* function from the *DODO Pair* smart contract
2. *DODO Pair* transfers the base tokens to the message sender
3. If the parameter *data* of the *buyBaseToken* function call is not null, the *DODO Pair* smart contract will call the *dodoCall* method of the message sender
4. After the *dodoCall* is executed, the *DODO Pair* smart contract will retrieve the quote tokens required for this transaction from the message sender

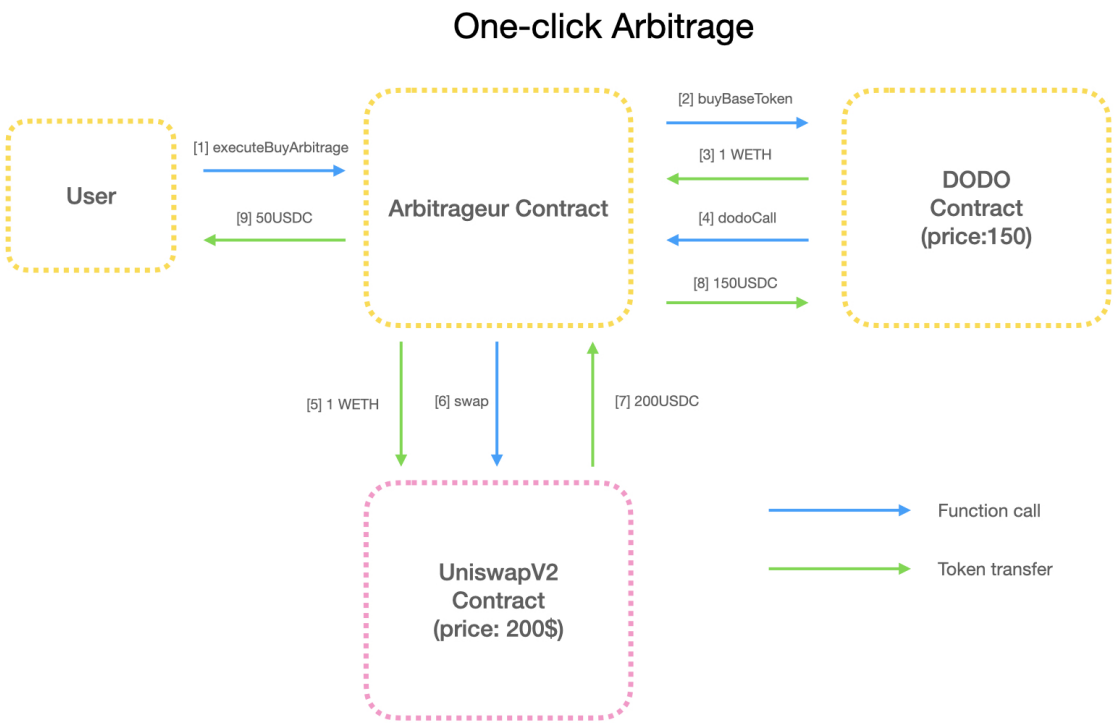
Please note: The *sellBaseToken* function can also perform flash swap in the same way.

4.1.3 What Can Flash Swap Do

Flash swap can significantly improve market efficiency. Market parity is maintained by arbitrageurs, and flash swap completely removes capital requirements for them, essentially eliminating the barrier of entry to arbitrage trading.

We will demonstrate a completely trustless and risk-free arbitrage trading contract as a use case of flash swap. Please refer to the *UniswapArbitrageur.sol* source code[4] for a concrete example. It has already been deployed and you can check out its Etherscan link here[5].

The following figure illustrates how an arbitrageur might take advantage of the price discrepancies between DODO and Uniswap.



A complete arbitrage trading maneuver consists of the following 9 steps:

1. The user calls *executeBuyArbitrage* on *UniswapArbitrageur*
2. *UniswapArbitrageur* calls *buyBaseToken* on *DODO Pair* and triggers flash swap
3. *DODO Pair* transfers 1 WETH to *UniswapArbitrageur*
4. *DODO Pair* calls *dodoCall* on *UniswapArbitrageur*
5. *UniswapArbitrageur* transfers 1 WETH received from *DODO Pair* to *UniswapV2*

6. *UniswapArbitrageur* calls swap on *UniswapV2*

7. *UniswapV2* transfers 200 USDC to *UniswapArbitrageur*

8. *DODO Pair* calls *transferFrom* and retrieves 150 USDC from *UniswapArbitrageur*

9. *UniswapArbitrageur* transfers the remaining 50 USDC to the user

In summary,

- Steps 2, 3, 4, and 8 take care of the DODO front
- Steps 5, 6, and 7 take care of the Uniswap front
- The user is only exposed to the process of sending transactions and making profits, with everything else abstracted away!

The best part about the *UniswapArbitrageur* contract is that users do not need any capital, nor do they need to know how DODO and Uniswap work. They would simply call a function and, if the execution succeeds, make a profit. If the execution fails, the users would only lose some gas.

In order to avoid unnecessary gas consumption, we recommend that users use *eth_call* to execute *executeBuyArbitrage* or *executeSellArbitrage* in advance to estimate arbitrage returns. If there is an arbitrage opportunity, these two functions will return profit of quote tokens and base tokens after successful execution.

4.1.4 Some Thoughts on Flash Swap

Once you have a deep understanding of flash swap, you will realize the superiority of the DeFi world over the centralized world. The composability of smart contracts has elevated the fund utilization of DeFi to an unprecedented level. Thanks to trustlessness, the cost of credit in DeFi is incredibly low. Once this financial system is integrated into the real world, its potential for improving our society and productivity will be truly boundless. The DODO team hopes that flash swap serves as a primer for DeFi builders and beginners alike to gain an appreciation for the power of DeFi.

Flash swap was inspired by dYdX and Uniswap. The DODO team genuinely appriciates and admires what these DeFi pioneers have done before us.

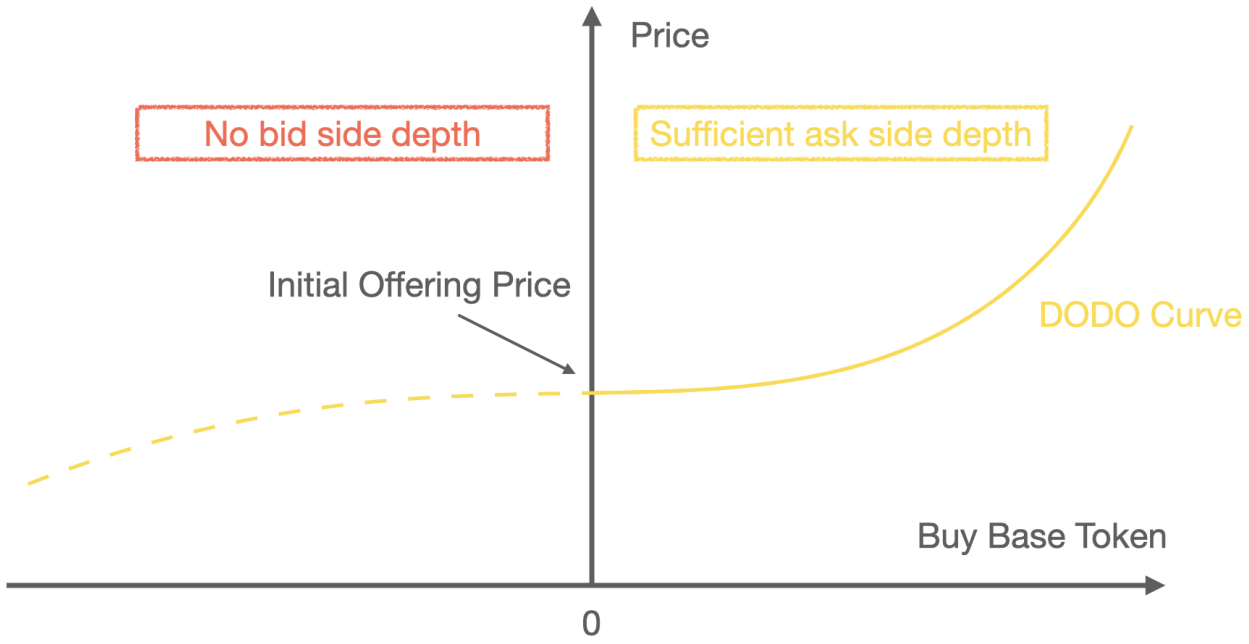
4.2 Initial DODO Offering

Initial DODO Offering (IDO) is a brand new approach to crypto asset issuance. Instead of paying exorbitant listing fees to get listed on CEXs or other DEXs, it is literally free to offer assets on DODO!

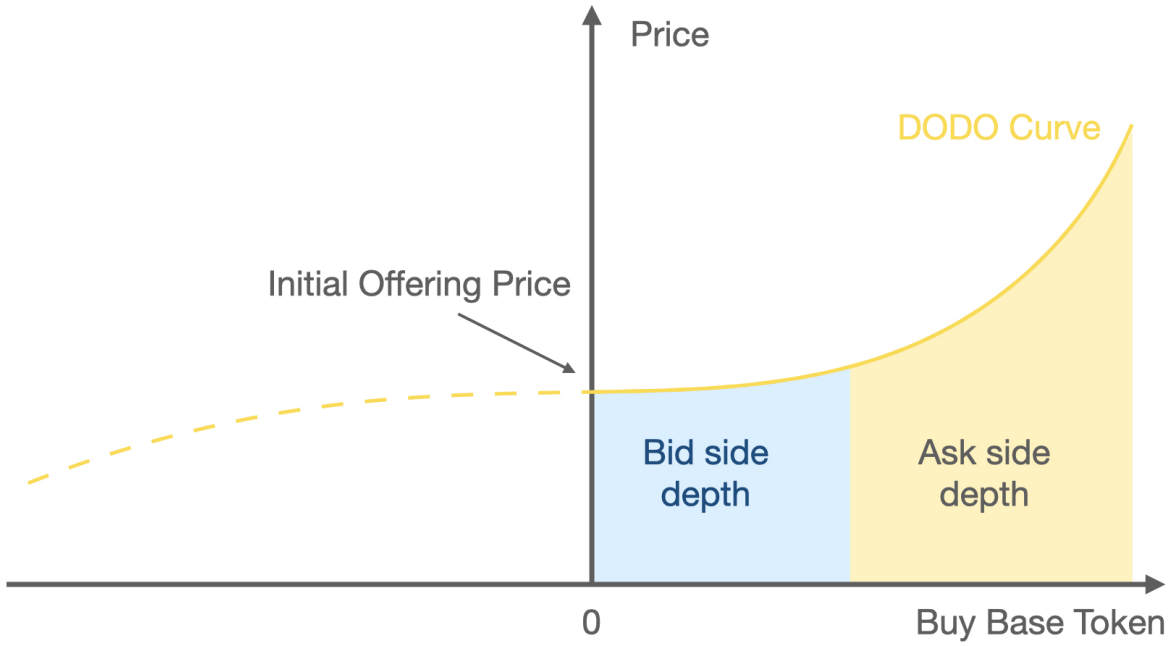
Normally, the PMM algorithm requires a price oracle to provide liquidity, but when there is no external market (which is usually the case when you are just starting your asset offering efforts), you can simply set the oracle price to a constant and start an initial DODO offering.

As discussed in previous sections, DODO, unlike AMM, does not require quote tokens. The only thing you need to do is to deposit your own tokens to the pool. After your token deposit, PMM creates ask side depth on its own. The more tokens you deposit, the better the liquidity.

Because there are no quote tokens in the pool, there is no bid side depth, but there is no need to worry. There are also no base tokens in the market and no one is selling either. IDO might feel somewhat similar to an auction, but there are some important differences.



Remember the constant price you set for the oracle? That price would be the initial offering price. When a trader buys your tokens, the price rises and quote tokens start flowing into the pool. These quote tokens then produce bid side depth as a result. Maybe we could call IDO a bidirectional auction.



Compared to AMM-based platforms, asset issuance on DODO provides more benefits:

- Sell tokens from an arbitrary price of your choice with zero capital requirement
- Sufficient and contract-fillable liquidity
- Flexible parameters (design your price curve by fine-tuning the parameters)

5 Authority

There are two special roles in each *DODO Pair* smart contract: *admin* and *supervisor*.

Here I would like to introduce the scope of power of *admin* and *supervisor*, and the principles of design behind.

5.1 Scope

Power of the supervisor is a subset of that of *admin*, and both *supervisor* and *admin* have A-level authority. Level A permissions include:

- Disable trade
- Disable deposit

- Set gas price limit

admin is the only one with B-level authority, which includes:

- Change admin
- Change supervisor
- Change maintainer
- Change oracle
- Set liquidity provider fee rate
- Set maintainer fee rate
- Set K
- Enable trade
- Enable deposit
- Final settlement

5.2 Principle

Level-A authority can be summarized as "freeze status" i.e. some functions of the system can be stopped urgently, but the status cannot be changed. In order to limit the power of *admin*, often actions taken by *admin* have to go through a complex governance process. To be risk resistant, we need a more flexible *supervisor* instead of an *admin* to take some actions that are not so sensitive but can significantly reduce system risks.

The B-level authority basically covers all aspects of the *DODO Pair* contract. The reason why so many parameters are designed to be variable is to better adapt to the rapidly changing market environment. It also leaves room for governance in the future.

It is worth pointing out that no one can prohibit users from withdrawing coins. Being non-custodial is the most important principle of Defi.

6 Decentralization

In our vision, DODO will be fully governed by the community, and controlled by three DAOs

- *Admin DAO*

Act as an administrator, the ultimate mediator of all issues.

- *Risk Control DAO*

Act as a supervisor and deal with all risk events urgently.

- *Earn DAO*

To distribute revenue of maintainer.

When DODO was launched, all authorities were governed by the team. As the community learns more about DODO, we will gradually return all the rights to the community. Although there is no timeline for this process yet, we do intend to follow the process.

Steps:

1. Set Admin to multi-sig wallet with daily limit
2. Deploy *DODOWild*: Allows anyone create their own DODO
3. Issue governance token
4. Set Maintainer to Earn DAO
5. Set Admin to Admin DAO
6. Set Supervisor to Risk Control DAO

What is the purpose of each step?

1. All admin actions come with a public announcement period to avoid single point failure
2. Anyone can create a new *DODO Pair* and use it to provide liquidity to their tokens. This marks the return of the code to the community
3. Issue governance tokens and formulate a token distribution plan, which will initiate the step down process
4. Hand over the profit distribution responsibility to Earn DAO
5. After handing admin authority over to the DAO, the team has no real control rights, and only reserves the right to control risk
6. The team steps down completely, marking the last step towards complete decentralization

7 Risk Parameters

7.1 Front Running

Front running on DODO could occur in the following scenario.

Arbitrageurs listen for oracle price updates transactions. If they see that the oracle price for an asset will go up in the next block, they will buy asset on DODO before the price update by paying higher gas prices. And sell the asset immediately after the oracle price has been updated. This will result in a loss for liquidity providers, and this loss is referred to as arbitrage loss.

This might seem like a big deal, but the truth is, such opportunities are few and far between, and not necessarily profitable for arbitrageurs.

First of all, front running is only profitable when the price fluctuates significantly. This is because DODO charges a 0.3% transaction fee per trade, thus buying and selling assets once incurs a 0.6% transaction fee overall. Therefore, if the price discrepancy between the oracle updates is less than 0.6%, front running is not profitable at all for arbitrageurs.

Secondly, DODO uses Chainlink as its oracle of choice. The Chainlink price oracle provides price updates by aggregating updates from 22 independent price feeders. This means that price changes on DODO are usually gradual and thus not susceptible to front running.

With that said, although the probability of a significant price change between updates is low, it will happen, and arbitrageurs looking to extract value from the system will take advantage of front running. The DODO team conducted extensive backtesting and discovered that in the overwhelming majority of cases, profit from market making far outweighs arbitrage loss for liquidity providers.

Please note that arbitrage loss due to front running will increase significantly during drastic market fluctuations. The DODO team recommends withdrawing your assets during fluctuations to avert the risk and proceeding with caution depending on your risk profile.

7.2 Fee Percentage

As mentioned above, the transaction fee deters arbitrageurs from extracting value from the system via front running, protecting liquidity providers from arbitrage loss.

The question is, what should the transaction fee percentage be? Lowering the percentage leads to more trading, potentially increasing profit for liquidity providers, but also elevating risk of arbitrage loss. On the other hand, increasing the percentage lowers the risk of arbitrage loss, but also reduces profit for liquidity providers. It is crucial to strike a reasonable balance between risk and profit.

Since market fluctuations have a bearing on arbitrageur behaviors, the fee percentage should also be fine-tuned based on market changes. The fee percentage should be low to facilitate more trading when the market is relatively stable, and high when the market is fluctuating. Determining the appropriate fee percentage is an important **governance** issue, and users should collectively have a say in how much risk they are willing to take on.

7.3 Parameter K

Another important parameter is k from the PMM pricing formula. A small k provides good liquidity and increase trading volume, but increases the risk of arbitrage loss; whereas a large k hurts liquidity and decreases trading volume, but reduces the risk of arbitrage loss. Therefore, similar to the fee percentage above, the value of k should be governed and determined by the users.

8 Backtest

8.1 Background

PMM stands for Proactive Market Maker, which is essentially a quantitative trading strategy used by liquidity providers (LP). To help LP understand ROI of PMM, we've performed a backtest to demonstrate the performance of PMM in different market environments.

8.2 Method

Evaluation of PMM focuses on these two aspects: profit and loss. The profit for LP is turnover rate multiplied by fee rate. While the loss has to be explained in two perspectives, counterparty risk and arbitrage trading. Counterparty risk can be ignored in this case, because PMM has built a mechanism to limit this risk. In addition, the risk comes from trades by normal users, which are almost random and are statistically balanced against. Arbitrage trading is inevitable and

contributes most of the loss, as onchain oracle price is always delayed from market. Hence, in the following backtesting, we focus on these two key values:

- Turnover rate (profit wise)
- Arbitrage loss (loss wise)

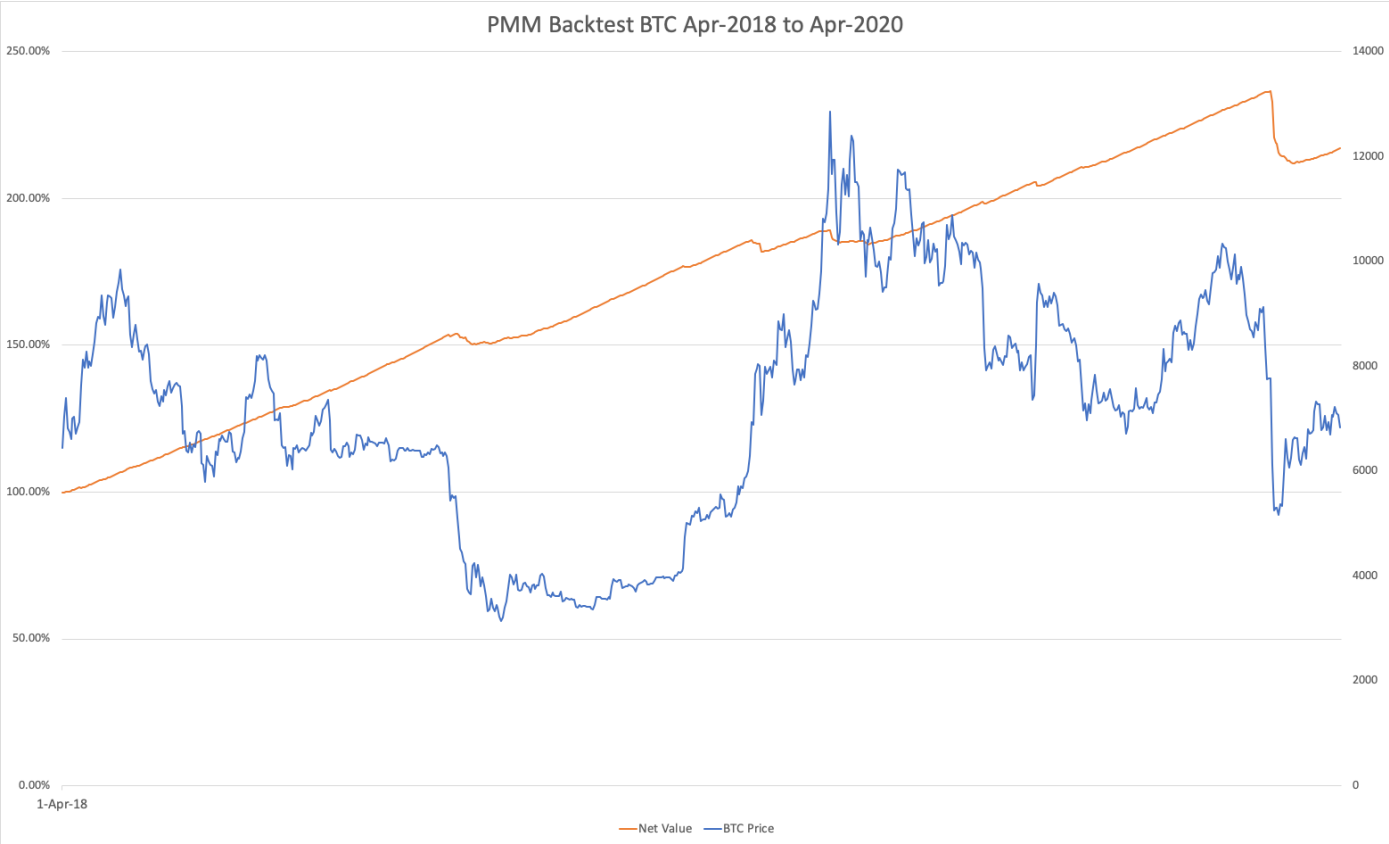
8.3 Profit Evaluation

Assumptions:

- Our pool size is 1/10 of uniswap's pool size
- Base Token and Quote Token have the same value
- PMM parameter k=0.1
- Fee rate 0.3%

Those assumptions are not set arbitrarily. Under this condition, PMM could provide the same liquidity as Uniswap, and hence it's reasonable to assume PMM has the same trading volume as Uniswap. However, because of aggregators, it's more realistic to assume PMM has half of the trading volume of Uniswap. According to history data[6], PMM daily turnover rate is about 100% and ROI is 0.3%.

Please note: The backtest report is written at 2020/7/19. We use Uniswap's historical data from 2020/6/1 to 2020/7/18.



8.4 Loss Evaluation

It's more complex to evaluate arbitrage loss, as no PMM-like algorithm has been deployed before. The best alternative is backtest with the most stringent standards. below is the assumptions:

- Onchain oracle price is always delayed from market price
- Oracle price updates whenever deviates from market price by more than 0.5% (chainlink threshold)
- Arbitrageurs always have enough funding and never miss a trade
- The external cost of arbitrageurs is 0.2% (including CEX fees and gas cost)

We backtested using BTC price from Apr-2018 to Apr-2020 with 1 minute interval. Aggregate profit and loss, we got the following conclusions.

8.5 Conclusion

The backtesting has covered most cases of the market environment, both the bull and bear market, even including the black swan event on 12th March. We concluded that:

- In most market environments, the fee income is sufficient to cover arbitrage losses and provides a very high rate of return (~80% APR)
- When the market changes volatilely, despite of rises or falls, LP will lose a significant amount of money

In brief, PMM makes profits when the market is flat, while makes losses when volatile.

8.6 Advantage & Disadvantage

Most quant strategies make profits only when market price goes up or down, and there is nothing to do when the market is flat. In contrast, PMM can make considerable profits when the price is nearly flat. Furthermore, unlike AMM, PMM never requires LP to deposit base and quote assets at a certain ratio. Instead, LP could deposit any amount of any asset as they want. As a result, PMM can be a supplement to the original strategies when the market is not volatile.

Nevertheless, we have to point out its disadvantages. As the old saying goes, there is no free lunch. When the market is volatile, LP suffers from significant loss. LP should make a balance between risks and benefits. So we recommend traders withdraw their assets when they predict the market to be volatile. As a decentralized project, what we can do is very limited. But we would definitely try our best to adjust system parameters to help LP, especially when black swan event happens.

In addition, one of the inherent drawbacks of backtesting is it cannot simulate 100% of the real trading. But to mitigate this risk, we have performed the backtesting with the most conservative assumptions. Still, LP should determine to what extent they trust the backtesting result.

8.7 FAQ about Backtest

1.Where does the turnover rate data come from?

We have counted the historical data of Uniswap in the past month. Because the capital utilization rate of PMM algorithm is very high, the capital utilization rate can reach ten times that of Uniswap. So the actual turnover rate is also much higher than Uniswap.

2.Why do you use BTC price for backtesting?

Because we did not find ETH price data of high-precision. We would be very grateful if someone could provide ETH price historical data with 1min interval or more frequent. But it is reasonable

to use BTC price to estimate loss, because ETH and BTC prices are highly correlated.

3.How does the arbitrage work?

The arbitrage is carried out when arbitrageurs notice that the price provided by the PMM is more beneficial to them than the market price, i.e. the difference between the PMM and the market price is less than its comprehensive arbitrage cost (PMM Fee + Arbitrage Cost)

4.Given that Chainlink's BTC Oracle accuracy rate is 1%, why is it set to 0.5% here?

First of all, Chainlink will increase the accuracy rate of BTC Oracle to 0.5% soon. Secondly, PMM will focus on ETH trading pair for now. And the accuracy rate of Chainlink's ETH Oracle is 0.5%.

5.Does the size of the funding pool have an impact on the backtest?

Yes, it does. ROI will not be so good if the pool size is too small. We need enough liquidity to compete with other liquidity sources. Actually, the 1/10 of Uniswap pool size required for backtesting is able to produce competitive liquidity, which equals only $900,000.

6.How about the gas cost

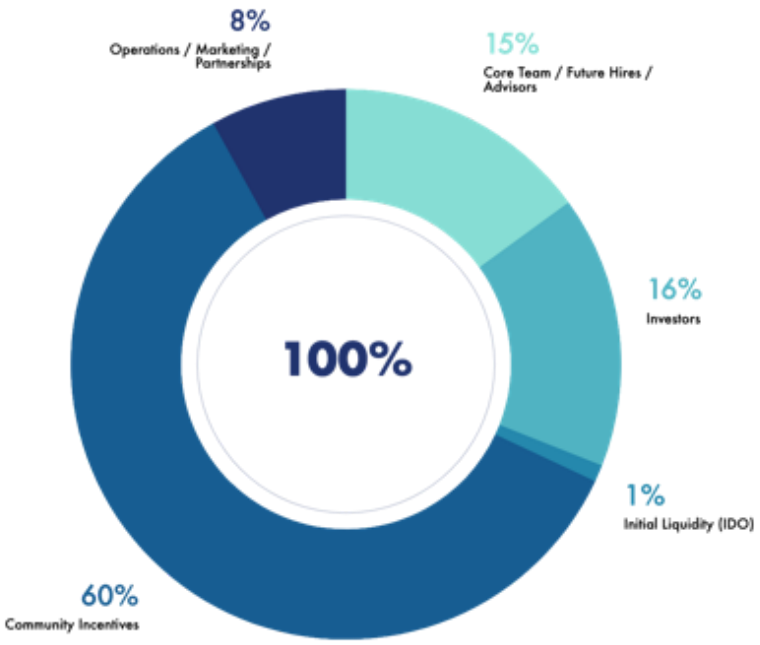
Swap between two standard ERC20 token cost 145,000 ~ 175,000 gas. The gas cost is slightly higher than Uniswap(~100,000 gas), but significantly lower than other protocols. For example, kyber costs ~400,000 gas; balancer costs ~300,000 gas; dydx costs ~400,000 gas;

9 Token Economy

Since inception, our goal at DODO Family has been to decentralize and diversify governance of the DODO Exchange. We would like to issue the Governance Token: DODO, to achieve this goal.

The total supply of DODO token is 1,000,000,000. The distribution would be:

DODO Token Distribution (Total Supply: 1 Billion)



60,000,000 DODO Token to Seed Round Investors. this part of token will be locked 1 year after token issuance, and then linearly vested over nest 2 years per Ethereum Block.

100,000,000 DODO Token to Private Round Investors. this part of token will be locked 6 month after token issuance, and then linearly vested over next 1 year per Ethereum Block.

150,000,000 DODO Token to Core Team / Future Hires / Advisors, this part of token will have the same lock up period with Seed Round Investors.

10,000,000 DODO Token reserved for Initial DODO Offering(IDO), this part of token will circulate immediately after IDO.

80,000,000 DODO Token reserved by DODOEX Foundation for operations, marketing campaign, partnership, exchange listing or future uses.

600,000,000 DODO Token reserved for community incentives. This part of Token will be distributed to DODO's supporter who participate in the protocol.

In DODO's vision for a prudent, truly decentralized governance model, individual traders and liquidity providers (LPs) assume essential roles in ensuring the integrity of the ecosystem as its

participants. The DODO team recognizes their importance in facilitating the growth of the DODO platform, and firmly believes that early adopters should be rewarded accordingly for their faith in the platform as it scales up. This is why the team intends to distribute DODO tokens to LPs in a fair and transparent manner.

Liquidity mining has been empirically proven by various DeFi projects to be an extremely effective and appealing way to incentivize participants, and the team may consider adopting this scheme going forward if needed. And the DODO Token mining and distribution strategy can be modified by DAO Governance in future.

References

[1]https://en.wikipedia.org/wiki/Bid%E2%80%93ask_spread

[2]https://github.com/DODOEX/dodo-smart-contract/blob/master/contracts/lib/DODOMath.sol

[3]https://github.com/DODOEX/dodo-smart-contract/blob/master/contracts/impl/Trader.sol

[4]https://github.com/DODOEX/dodo-smart-contract/blob/master/contracts/helper/UniswapArbitrageur.sol

[5]https://etherscan.io/address/0xbf90b54cc00ceeaa93db1f6a54a01e3fe9ed4422

[6]https://info.uniswap.org/pair/0xb4e16d0168e52d35cacd2c6185b44281ec28c9dc