SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols

Jiahua Xu
UCL CBT

Nazariy Vavryk
reNFT

Krzysztof Paruch
Vienna University of Economics and Business

Simon Cousaert
UCL CBT

***Abstract*—As an integral part of the Decentralized Finance (DeFi) ecosystem, Automated Market Maker (AMM) based Decentralized Exchanges (DEXs) have gained massive traction with the revived interest in blockchain and distributed ledger technology in general. Most prominently, the top six AMMs—Uniswap, Balancer, Curve, DODO, Bancor and Sushiswap—hold in aggregate 15 billion USD worth of crypto-assets as of March 2021. Instead of matching the buy and sell sides, AMMs employ a peer-to-pool method and determine asset price algorithmically through a so-called conservation function. Compared to centralized exchanges, AMMs exhibit the apparent advantage of decentralization, automation and continuous liquidity. Nonetheless, AMMs typically feature drawbacks such as high slippage for traders and divergence loss for liquidity providers. This work establishes a general AMM framework describing the economics and formalizing the system's state-space representation. We employ our framework to systematically compare the top AMM protocols' mechanics, deriving their slippage and divergence loss functions. We further discuss security and privacy concerns associated with AMM DEXs, and conduct a comprehensive literature review on related work covering both DeFi and conventional market microstructure.**

***Index Terms*—Decentralized Finance, decentralized exchange, automated market maker, blockchain, Ethereum**

I. Introduction

A. Background

With the revived interest in blockchain and cryptocurrency among both the general populace and institutional actors, the past year has witnessed a surge in crypto trading activity and increasing competition and accelerated development in the Decentralized Finance (DeFi) space.

Among all the prominent DeFi applications, Automated Market Makers (AMM) based Decentralized Exchanges (DEXs) are on the ascendancy, with an aggregate value locked exceeding 15 billion USD at the time of writing.[1] Different from order-book based exchanges where the market price of an asset is determined by the last matched buy and sell orders, each AMM uses a so-called conservation function that determines asset price algorithmically by only allowing the exchange rates to move along predefined trajectories which are conditioned upon the quantities of available assets. AMMs implement a peer-to-pool method, where liquidity providers contribute assets to liquidity pools while individual users exchange assets with a pool containing both the input and the output assets. Exchange users obtain immediate liquidity without having to find an exchange counterparty first, whereas liquidity providers benefit from asset supply with exchange fees from pool users. Furthermore, maintaining the state of an order book is computationally expensive, which is costly given the native price mechanism on the Ethereum blockchain. While this problem is minimized by keeping the order books off-chain, DEX with AMMs allows for more accessible liquidity provision, especially for low-liquid assets.

Despite apparent advantages such as decentralization, automation and continuous liquidity, AMMs are often characterized by high slippage with asset exchange and divergence loss with liquidity provision. Throughout the last three years, new protocols have been introduced to the market one after another with incremental improvements and the attempt to tackle different issues which were identified as weak spots in a previous version.

While innovative on certain aspects, the various AMM protocols generally consist of the same set of composed mechanisms to allow for multiple functionalities of the system. Therefore these systems are structurally similar, and their main differences lie in parameter choices and/or mechanism adaptations. Describing a **class of mechanisms** that defines the characteristics of an AMM allows assessing the differences of design choices and their impacts on the system to consequently discuss the structural composition of AMMs and make statements about their stability for different market conditions.

B. Contributions

This work establishes a taxonomy of the major components of an AMM focusing on stakeholder roles, asset types, mechanisms, metrics and attack vectors. This framework is used for a comparative analysis of selected projects by comparing slippage and divergent loss functions of example protocols.

This work represents the first systematization of knowledge in AMM-based DEXs with deployed protocol examples to the best of our knowledge.

II. AMM Preliminaries

This section presents a taxonomy of the main components across major decentralized exchanges [1]. To guide the following formal definitions, Uniswap can be used as an intuitive example. This protocol allows exchanging two tokens via the use of one liquidity pool containing both assets. A constant product function is parametrized at the time of pool inception, defining a number that has to hold true as the product of both asset quantities for all future states of the system. This property

[1] https://defipulse.com/

determines the swap prices as any trade must uphold the constant product under updated asset amounts in the pool. The liquidity is provided to the pool by liquidity providers, which receive a pool share representing their relative contribution to the pool. The trading fee for token swaps is accumulated in the pool and therefore acts as a reward for liquidity providers. Other protocols extend this basic functioning, and their specific components are discussed later in this paper.

A. Actors

a) Liquidity provider (LP): A liquidity pool creator is the first liquidity provider (LP) when deploying a new smart contract that acts as a liquidity pool with some initial supply of crypto assets. Other LPs can subsequently increase the pool's reserve by adding more of the assets that are contained in the pool. In turn, they receive pool shares proportionate to their liquidity contribution as a fraction of the entire pool [2]. LPs earn transaction fees paid by exchange users. While sometimes subject to a withdrawal penalty, LPs can freely remove funds from the pool [3] by surrendering a corresponding amount of pool shares [2].

The liquidity pool must be initialized with two or more different assets for the smart contract to parametrize the conservation function and set the initial relative prices. Pool creators initialize the pool with quantities that reflect the market prices to avoid unnecessary divergence loss. The act of liquidity provision or removal updates the value of the conservation function invariant(s) (see Section III). Liquidity providers are also called *liquidity miners* due to new protocol tokens minted and distributed to them as a reward in addition to pool shares when they supply funds. Like centralized exchanges, an AMM-based DEX can facilitate an initial exchange offering to supply a new asset through liquidity pool creation.

b) Exchange user (Trader): A trader submits an exchange order to a liquidity pool by specifying the input and output asset and one associated quantity - the smart contract will automatically calculate the exchange rate based on the conservation function and execute the exchange order accordingly. A fee is charged on top of each trade to compensate liquidity providers for their capital provision.

c) Arbitrageurs: Arbitrageurs are a particular type of exchange users who compare asset prices across different markets to execute trades whenever closing price gaps can extract profits. In doing so, arbitrageurs ensure consistency of asset price in other decentralized, centralized, on-chain and off-chain exchanges.

d) Protocol foundation: Protocol foundation consists of protocol founders, designers, and developers responsible for designing and improving the protocol. The development activities are often funded directly or indirectly through accrued earnings such that the foundation members are financially incentivized to build a user-friendly protocol that can attract high trading volume.

B. Assets

Several distinct sorts of assets are used in AMM protocols for operations and governance. One or more assets can fulfil several functionalities; one asset may assume multiple roles.

a) Risk assets: This is the primary type of asset for which the protocol was designed: to provide liquidity in these assets, to facilitate exchange between them and to allow liquidity providers to earn rewards in return for their contribution. Typically many different risk assets are involved in one protocol - they have to be whitelisted, compatible with the protocol and fulfil the technical requirements (e.g. ERC20[2] for most AMMs on Ethereum).

b) Base assets: For some protocols, a trading pair always consists of a risk asset and a designated base asset. In the case of Bancor, every risk asset is paired with `BNT`, the protocol's native token with an elastic supply [4]. In their early version, Uniswap required every pool to be initiated with `ETH` as one of the risk assets making it an obligatory base asset. Many protocols, such as Balancer and Curve are managed without a designated base asset as they connect two or more risk assets directly in the composition of their portfolios.

c) Pool shares: Also known as "liquidity shares" and "liquidity provider shares", pool shares represent ownership in the portfolio of assets within a pool, and are distributed to liquidity providers. Shares qualify for the reception of fees that are earned in the portfolio whenever a trade occurs. Shares can be redeemed at any time to withdraw back the liquidity initially provided.

d) Protocol tokens: Protocol tokens are used to represent voting rights in a decision formation process defined in the protocol and are thus also termed "governance tokens". Protocol tokens are typically valuable assets that are sometimes tradeable even outside of the AMM and can incentivize participation. For example, they might be rewarded to liquidity providers in proportion to their liquidity supply.

C. Fundamental AMM economics

1) Rewards: AMM protocols often run several reward schemes, including liquidity reward, staking reward, governance rights, and security reward, distributed to different actors to encourage participation and contribution.

a) Liquidity reward: Liquidity providers are rewarded for supplying assets to a liquidity pool, as this can be deemed service for the broader community for which they have to bear the opportunity costs associated with funds being locked in the pool. Liquidity providers receive their share of trading fees paid by exchange users.

b) Staking reward: On top of the liquidity reward in the form of transaction income, liquidity providers are offered the possibility to stake certain tokens as part of an initial incentive program from the token protocol. The ultimate goal of the individual token protocols (see e.g. `GIV` [5] and `TRIPS` [6]) is to further encourage token holding, while simultaneously facilitating token liquidity.

[2]https://eips.ethereum.org/EIPS/eip-20

c) Governance right: An AMM may encourage liquidity provision and/or swapping by rewarding participants governance right in the form of protocol tokens (see II-B). AMMs compete with each other to attract funds and trading volume. To bootstrap an AMM in the early phase with incentivized early pool establishment and trading, a feature called liquidity mining can be installed where the native protocol's tokens are minted and issued to liquidity providers and/or exchange users.

d) Security reward: Just as every protocol built on top of an open, distributed network, AMM-based DEXs on Ethereum suffer from security vulnerabilities. Besides code auditing, a common practice that a protocol foundation adopts is to have the code vetted by a broader developer community and reward those who discover and/or fix securities of the protocol with monetary prizes, commonly in fiat currencies, through a bounty program.

2) Explicit costs: Interacting with AMM protocols incurs various costs, including charges for some form of "value" created or "service" performed and fees for interacting with the blockchain network. AMM participants need to anticipate three types of fees: liquidity withdrawal penalty, swap fee and gas fee.

a) Liquidity withdrawal penalty: As introduced in III-B and demonstrated in Section IV later in this paper, withdrawal of liquidity changes the shape of the conservation function and negatively affects the usability of the pool by elevating the slippage. Therefore, AMMs such as DODO [7] levy a liquidity withdrawal penalty to discourage this action.

b) Swap fee: Users interacting with the liquidity pool for token exchanges have to reimburse liquidity providers for the supply of assets. This compensation comes in the form of swap fees that are charged in every exchange trade and then distributed to liquidity pool shareholders. A small percentage of the swap fees may also go to the foundation of the AMM to further develop the protocol.

c) Gas fee: Every interaction with the protocol is executed in the form of an on-chain transaction and is thus subject to gas fee applicable to all transactions on the underlying blockchain. In a decentralized network validating nodes verifying transactions need to be compensated for their efforts, and transaction initiators must cover these operating costs. The paid gas fees depend on the price of, for example, `ETH` and the gas price of the transaction chosen by the user. The average gas price evolution shows that the gas fees are becoming increasingly more substantial[3] as a result of the growing adoption of Ethereum. Compounded with the rising `ETH` price and the complexity of AMM contracts, gas fees must be taken into consideration when interacting with AMMs.

3) Implicit costs: Two essential implicit costs native to AMM-based DEXs are slippage for exchange users and divergence loss for liquidity providers.

a) Slippage: Slippage is defined as the difference between the spot price and the realized price of a trade and is caused by the curve design of an AMM that dictates the asset prices. Instead of matching buy and sell orders, exchange rates are determined on a continuous curve. Every trade on an AMM-based DEX will *always* encounter slippage conditioned upon the trade size, the pool amounts and the exact design of the conservation function. The spot price approaches the realized price for infinitesimally small trades, but they deviate more for bigger trade sizes. This effect is amplified for smaller liquidity pools as every trade will significantly impact the relative quantities of assets in the pool and, therefore, higher slippage.

b) Divergence loss: For liquidity providers, assets supplied to a protocol are still exposed to volatility risk, which comes into play in addition to the loss of time value of locked funds. A swap alters the asset composition of a pool, which automatically updates the asset prices implied by the conservation function of the pool (Equation 3). This consequently changes the value of the entire pool. Compared to holding the assets outside of an AMM pool, contributing the same amount of assets to the pool in return for pool shares can result in less value with price movement, an effect termed "divergence loss" or "impermanent loss" (see Section IV). This loss is "impermanent" because as asset price moves back and forth, the depreciation of the pool value disappears and reappears all the time and is only realized when assets are actually taken out of the pool.

Since assets are bonded together in an pool, changes of prices in one asset affect all others in this pool. For an AMM protocol that supports single-asset supply, this forces liquidity providers to be exposed to risk assets they have not been holding in the first place.

III. Formalization of Mechanisms

Overall, the functionality of an AMM can be generalized formally by a set of few mechanisms. These mechanisms define how users can interact with the protocol and what the response of the protocol will be given particular user actions. Action-related mechanisms dictate how *providing and withdrawing liquidity* as well as *swapping assets* are executed where protocol-related properties define how *fees and rewards* are calculated. On top of that, there are some protocol-specific mechanisms for *governance* and *security*, but the aforementioned basic mechanisms are the same.

While all AMMs are similar in this basic functionality, they have two ways of differentiating themselves and propose improvements to existing protocols. The first option is to take an existing implementation as given, copy and reuse all mechanisms and adjust the fee and reward structure to change the incentives and payouts of the protocol and by doing so, potentially improve targeted user or protocol metrics. The second option is to change the composition or functionality of the mechanism or propose a new pool structure that will call for major changes in how the protocol works. In this way, key metrics can be improved as well but in a completely different way.

[3]https://ycharts.com/indicators/ethereum_average_gas_price

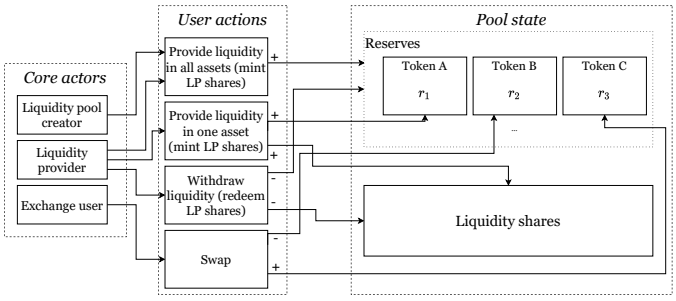


Figure 1: Stylized AMM mechanism

A. State space representation

The functioning of any blockchain-based system can be modeled using the state-space terminology. States and agents constitute main system components; protocol activities are described as actions (Figure 1); the evolution of the system over time is modelled with state transition functions. This can be generalized into a state transition function f encoded in the protocol such that $\chi \xrightarrow[f]{a} \chi'$, where $a \in A$ represents an action imposed on the system while χ and χ' represent the current and future states of the system respectively.

The object of interest is the state χ of the liquidity pool which can be described with

$$\chi = (\{r'_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{C}, \Omega) \tag{1}$$

where r_k denotes the quantity of token k in the pool, p_k the current spot price of token k, $\mathcal{C}$ the conservation function invariant(s), and Ω the collection of protocol hyperparameters. This formalization can encompass various AMM designs.

The most critical design component of an AMM is its conservation function which defines the relationship between different state variables and the invariant(s) $\mathcal{C}$. The conservation function is protocol-specific as each protocol seeks to prioritize a distinct feature and target particular functionalities. For example, Uniswap implements an elementary conservation function that achieves a low gas fee; Balancer's conservation function links mul whereas Curve's conservation function is complex but guarantees low slippage (see Section IV).

The core of an AMM system state is the quantity of each asset held in a liquidity pool. Their sums or products are typical candidates for invariants. Examples of a constant-sum market maker include mStable [8]. Uniswap [9] represents constant-product market makers, while Balancer [3] generalizes this idea to a geometric mean. The Curve [10] conservation function is notably a combination of constant-sum and constant-product (see Section IV).

B. Liquidity change and asset swap

Hyperparameter set Ω is determined at pool creation and shall remain the same afterwards. While this value of hyperparameters might be changed through protocol governance activities, this does not and should not occur on a frequent basis.

Invariant $\mathcal{C}$, despite its name, refers to the pool variable that stays constant only with swap actions but changes at liquidity provision and withdrawal. In contrast, trading moves the price of traded assets; specifically, it increases the price of the output asset relative to the input asset, reflecting a value appreciation of the output asset driven by demand. Liquidity provision and withdrawal, on the other hand, should not move the asset price.

General rules of AMM-based DEX

1) The price of assets in an AMM pool stays constant for *pure* liquidity provision and withdrawal activities.
2) The invariant of an AMM pool stays constant for *pure* swapping activities.

Formally, the state transition induced by *pure* liquidity change and asset swap can be expressed as follows.

$$\begin{aligned}&(\{r_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{C}, \Omega)\\ \xrightarrow[f]{\text{liquidity change}}&(\{r'_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{C}', \Omega)\end{aligned} \tag{2}$$

$$\begin{aligned}&(\{r_k\}_{k=1,\dots,n}, \{p_k\}_{k=1,\dots,n}, \mathcal{C}, \Omega)\\ \xrightarrow[f]{\text{swap}}&(\{r'_k\}_{k=1,\dots,n}, \{p'_k\}_{k=1,\dots,n}, \mathcal{C}, \Omega)\end{aligned} \tag{3}$$

Note that protocol-specific intricacies may result in asset price change with liquidity provision/withdrawal, or invariant $\mathcal{C}$ update with trading.

The asset spot price can remain the same only when assets are added to or removed from a pool proportionate to the current reserve ratio ($r_1 : r_2 : \dots : r_n$). Disproportionate addition or removal can be treated as a combination of two actions: proportionate reserve change plus asset swap, such as with Balancer [3]. Therefore, this action is no longer a *pure* liquidity provision/withdrawal and would thus move the asset spot price.

Specific fee mechanisms also cause invariant $\mathcal{C}$ to become variant through trading. Specifically, when trading fees are kept within the liquidity pool, a trading action can be decomposed into asset swap and liquidity provision. This action is, therefore, no longer a *pure* asset swap and would thus move the value of $\mathcal{C}$ (see e.g. [11]). Also, as float numbers are not yet fully supported by Solidity [12]—the language for Ethereum smart contracts, AMM protocols typically recalculate invariant $\mathcal{C}$ after each trade to minimize the accumulation of rounding errors.

C. Generalized formulas

In this section, we generalize AMM formulas necessary for demonstrating the interdependence between various AMM state variable, as well as for computing slippage as well as divergence loss. Mathematical notations and their definitions can be found in Table I.

Table I: Mathematical notations for pool mechanisms

Notation	Definition	Applicable protocols
Preset hyperparameters, Ω		
w_k	Weight of asset reserve r_k	Balancer
$\mathcal{A}$	Slippage controller	Uniswap V3, Curve, DODO
State variables		
$\mathcal{C}$	Conservation function invariant	all
r_k	Quantity of token$_k$ in the pool	all
p_k	Current spot price of token$_k$	all
Process variables		
x_i	Input quantity added to reserve of token$_i$ (removed when $x_i < 0$)	all
ρ	Token value change	all
Functions		
C	Conservation function	all
Z	Implied conservation function	all
${}_iE_o$	token$_o$ price denominated in token$_i$	all
S	Slippage	all
V	Reserve value	all
L	Divergence loss	Uniswap, Balancer, Curve

1) Conservation function: An AMM conservation function, also termed "bonding curve", can be expressed explicitly as a relational function between AMM invariant and reserve quantities $\{r_k\}_{k=1,\dots,n}$:

$$\mathcal{C} = C(\{r_k\}) \tag{4}$$

A conservation function for each token pair, say r_i—r_0, must be concave, nonnegative and nondecreasing [13] (see also Figure 3). For complex AMMs such as Curve, it might be convenient to express the conservation function implicitly in order to derive exchange rates between two assets in a pool:

$$Z(\{r_k\};\mathcal{C}) = C(\{r_k\}) - \mathcal{C} = 0 \tag{5}$$

2) Spot exchange rate: The spot exchange rate between token$_i$ and token$_o$ can be calculated as the slope of the r_i—r_o curve (see examples in Figure 3) using partial derivatives of the conservation function Z.

$${}_iE_o(\{r_k\};\mathcal{C}) = \frac{\partial Z(\{r_k\};\mathcal{C})/\partial r_o}{\partial Z(\{r_k\};\mathcal{C})/\partial r_i} \tag{6}$$

Note that ${}_iE_o = 1$ when $i = o$.

3) Swap amount: The amount of token$_o$ received x_o (spent when $x_o < 0$) given amount of token$_i$ spent x_i (received when $x_i < 0$) can be calculated following the steps below.

a) Update reserve quantities: Input quantity x_i is simply added to the existing reserve of token$_i$; the reserve quantity of any token other than token$_i$ or token$_o$ stays the same:

$$r_i' \coloneqq R_i(x_i; r_i) = r_i + x_i \tag{7}$$

$$r_j' = r_j, \qquad \forall j \neq i, o \tag{8}$$

b) Compute new reserve quantity of token$_o$: The new reserve quantity of all tokens except for token$_o$ is known from the previous step. One can thus solve r_o', the unknown quantity of token$_o$, by plugging it in the conservation function:

$$Z(\{r_k'\};\mathcal{C}) = 0 \tag{9}$$

Apparently, r_o' can be expressed as a function of the original reserve composition $\{r_k\}$, input quantity x_i, namely,

$$r_o' \coloneqq R_o(x_i, \{r_k\};\mathcal{C}) \tag{10}$$

c) Compute swapped quantity: The quantity of token$_o$ swapped is simply the difference between the old and new reserve quantities:

$$x_o \coloneqq X_o(x_i, \{r_k\};\mathcal{C}) = r_o - r_o' \tag{11}$$

4) Slippage: Slippage measures the deviation between effective exchange rate $\frac{x_i}{x_o}$ and the pre-swap spot exchange rate ${}_iE_o$, expressed as:

$$S(x_i, \{r_k\};\mathcal{C}) = \frac{x_i/x_o}{{}_iE_o} - 1 \tag{12}$$

5) Divergence loss: Divergence loss describes the loss in value of the all reserves in the pool compared to holding the reserves outside of the pool, after a price change of an asset. Based on the formulas for spot price and swap quantity established above, the divergence loss can generally be computed following the steps described below. In the valuation, we assign token$_i$ as the denominating currency for all valuations. While the method to be presented can be used for multiple token price changes through iterations, we only demonstrate the case where only the value of token$_o$ increases by ρ, while all other tokens' value stay the same. Token$_i$ is the numéraire. Designating one of the tokens in the pool as a numéraire can also be found in DeFi simulation papers such as [13].

a) Calculate the original pool value: The value of the pool denominated in token$_i$ can be calculated as the sum of the value of all token reserves in the pool, each equal to the reserve quantity multiplied by the exchange rate with token$_i$:

$$V(\{r_k\};\mathcal{C}) = \sum_j {}_iE_j(\{r_k\};\mathcal{C}) \cdot r_j \tag{13}$$

b) Calculate the reserve value if held outside of the pool: If all the asset reserves are held outside of the pool, then a change of ρ in token$_o$'s value would result in a change of ρ in token$_o$ reserve's value:

$$V_{\text{held}}(\rho; \{r_k\}, \mathcal{C}) = V(\{r_k\};\mathcal{C}) + [{}_jE_o(\{r_k\};\mathcal{C}) \cdot r_o] \cdot \rho$$

c) Obtain re-balanced reserve quantities: Exchange users and arbitrageurs constantly re-balance the pool through trading in relatively "cheap", depreciating tokens for relatively "expensive", appreciating ones. As such, asset value movements are reflected in exchange rate changes implied by the dynamic pool composition.

Therefore, the exchange rate between token$_o$ and each other token$_j$ ($j \neq o$) implied by new reserve quantities $\{r_k'\}$, compared to that by the original quantities $\{r_k\}$, must satisfy equation sets 14. At the same time, the equation for the conservation function must stand (Equation 15).

$$\rho = \frac{{}_jE_o(\{r_k'\};\mathcal{C})}{{}_jE_o(\{r_k\};\mathcal{C})} - 1, \qquad \forall j \neq o \tag{14}$$

$$0 = Z(\{r_k'\};\mathcal{C}) \tag{15}$$

Table II: Comparison Table of discussed DEXs: value locked, trade volume of the past 7 days, the market share by the last 30 days volume, the governance token, the number of governance token holders and the fully diluted value, as on 15 April 2021. Data retrieved from DeFi Pulse and Dune Analytics on 22 March 2021.

Protocol	Value locked (billion USD)	Trade volume (billion USD)	Market (%)	Governance token	Governance token holders	Fully diluted value (billion USD)
Uniswap	6.30	6.99	53.8	UNI	212,506	37.4
Sushiswap	4.26	1.99	15.4	SUSHI	43,306	3.78
Curve	5.33	1.89	14.6	CRV	28,669	4.89
Bancor	1.96	0.66	5.1	BNT	38,881	1.37
Balancer	2.33	0.46	3.6	BAL	34,225	2.8
DODO	0.07	0.15	1.16	DODO	9,568	5.0

A total number of n-equations ($n-1$ with equation sets 14, plus 1 with Equation 15) would suffice to solve n unknown variables $\{r'_k\}_{k=1,\dots,n}$, each of which can be expressed as a function of ρ and $\{r_k\}$:

$$r'_k := R_k(\rho, \{r_k\}; \mathcal{C}) \tag{16}$$

d) Calculate the new pool value: The new value of the pool can be calculated by summing the products of the new reserve quantity multiplied by the new price (denominated by token_i) of each token in the pool:

$$V'(\rho, \{r_k\}; \mathcal{C}) = \sum_j {}_iE_j(\{r'_k\}; \mathcal{C}) \cdot r'_j \tag{17}$$

e) Calculate the divergence loss: Divergence loss can be expressed as a function of ρ, the change in value of an asset in the pool:

$$L(\rho, \{r_k\}; \mathcal{C}) = \frac{V'(\rho, \{r_k\}; \mathcal{C})}{V_{\text{held}}(\rho; \{r_k\}, \mathcal{C})} - 1 \tag{18}$$

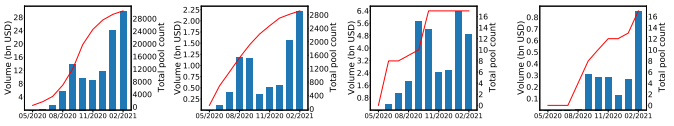
IV. Comparison of AMM protocols

AMM-based DEXs are home to billions of dollars worth of on-chain liquidity. Table II lists major AMM protocols, their respective value locked, as well as some other general metrics. Uniswap is undeniably the biggest AMM measured by trade volume, as confirmed by the volume growth displayed in Figure 2a, and the number of governance token holders, although it is remarkable that Sushiswap has more value locked within the protocol. The number of governance token holders of smaller protocols as Bancor and Balancer is relatively significant compared to Curve token holders, as they do approximately half of the volume but have 25 to 50% more governance token holders.

A. Major AMM protocols

This section focuses on the four most representative AMMs: Uniswap (including V2 and V3), Balancer V1, Curve, and DODO. Figure 2 shows an increasing trend over those AMMs. While Curve only has 17 pools at the moment of writing, Uniswap has over 30,000. DODO is relatively new but already showcases around 200% growth compared to the start of 2021 in terms of volume.

We describe the liquidity pool structures of those protocols in the main text. We also derive the conservation function, slippage, as well as divergence loss of those protocols. A summary of formulas can be found in Table III. We refer our readers to Appendix A for a detailed explanation and derivation of those formulas. The protocols' conservation function, slippage, as well as divergence loss under different hyperparameter values are plotted in in Figure 3, Figure 4 and Figure 5, respectively. We always use token_1 as price or value unit; namely, token_1 is the assumed numéraire.



(a) Uniswap (b) Balancer (c) Curve (d) DODO

Figure 2: Monthly volume and the total pool count for AMMs. Data from The Graph and Dune Analytics.

1) Uniswap V2: The Uniswap protocol prescribes that a liquidity pool always consists of one pair of assets. The pool's smart contract always assumes that the reserves of the two assets have equal value. Uniswap implements a conservation function with a constant-product invariant.

2) Uniswap V3: Uniswap V3 enhances Uniswap V2 by allowing liquidity provision to be concentrated on a fraction of the bonding curve, thus virtually amplifying the conservation function invariant and reducing the slippage. Uniswap V3 is a special case of the Uniswap V2 with the slippage controller $\mathcal{A} \to \infty$ (Figure 3a).

3) Balancer: The Balancer protocol allows each liquidity pool to have more than two assets [3]. Each asset reserve r_k is assigned with a weight w_k at pool creation, where $\sum_k w_k = 1$. Weights are pool hyperparameters and do not change with either liquidity provision/removal or asset swap. The weight of an asset reserve represents the value of the reserve as a fraction of the pool value. Balancer can also be deemed a generalization of Uniswap; the latter is a special case of the former with $w_1 = w_2 = \frac{1}{2}$ (Figure 3b).

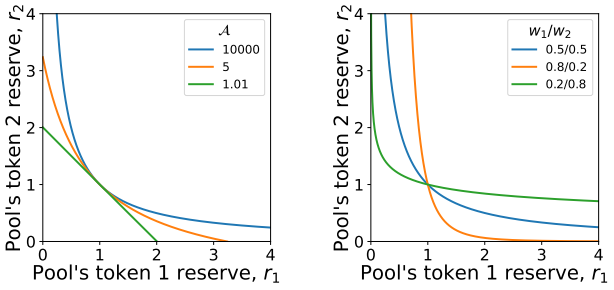
4) Curve: With the Curve protocol, formerly StableSwap [10], a liquidity pool consists of two or more assets with the same peg, for example, `USDC` and `DAI`, or `wBTC` and `renBTC`. Curve approximates Uniswap V2 when its constant-sum component has a near-0 weight (Figure 3c).

5) DODO: DODO only supports 2-asset liquidity pools at the moment. The pool creator opens a new pool with some reserves on both sides, which determines the pool's initial equilibrium state.

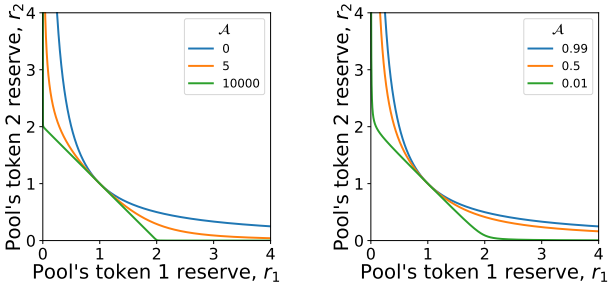
Notably, the pool permanently anchors the exchange rate between the two assets to the external data fed by a price oracle. Thus, unlike Uniswap, where their reserve quantities imply asset exchange rates, DODO allows the reserve ratio between the two assets within a pool to be arbitrary while still maintaining the asset exchange rate close to the market rate. Due to this feature, DODO differentiates itself from traditional AMMs and terms their pricing algorithm as the "Proactive Market Making" algorithm, or PMM.

Table III: Function comparison table of Uniswap, Balancer, Curve and DODO. Formulas are derived in Appendix A. Conservation functions are visualized in Figure 3, slippage functions in Figure 4 and divergence loss functions in Figure 5.

	Uniswap V2	**Uniswap V3**	**Balancer**	**Curve**	**DODO**
Conservation function $Z(\{r_k\};\mathcal{C})=0$	$r_1 \cdot r_2$	$\left(r_1+\frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}\right)\cdot\left(r_2+\frac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1}\right)$ $=\frac{\mathcal{A}\cdot\mathcal{C}_1\cdot\mathcal{C}_2}{(\sqrt{\mathcal{A}}-1)^2}$	$\mathcal{C}=\prod_k r_k^{w_k}$	$\mathcal{A}\left(\frac{\sum_k r_k}{\mathcal{C}}-1\right)=\frac{\left(\frac{\mathcal{C}}{n}\right)^n}{\prod_k r_k}-1$	$\begin{cases} r_1-\mathcal{C}_1=P\cdot(\mathcal{C}_2-r_2)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_2}{r_2}-1\right)\right], & r_1\geq\mathcal{C}_1 \\ r_2-\mathcal{C}_2=\frac{(\mathcal{C}_1-r_1)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_1}{r_1}-1\right)\right]}{P}, & r_1\leq\mathcal{C}_1 \end{cases}$
Spot Exchange rate $={}_iEo(\{r_k\};\mathcal{C})=\frac{\partial Z(\{r_k\};\mathcal{C})/\partial r_o}{\partial Z(\{r_k\};\mathcal{C})/\partial r_i}$	$\frac{r_1}{r_2}$	$\frac{r_1+\frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}}{r_2+\frac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1}}$	$\frac{r_1\cdot w_2}{r_2\cdot w_1}$	$\frac{r_1\cdot\left[\mathcal{A}\cdot r_2\cdot\prod_k r_k+\mathcal{C}\cdot\left(\frac{\mathcal{C}}{n}\right)^n\right]}{r_2\cdot\left[\mathcal{A}\cdot r_1\cdot\prod_k r_k+\mathcal{C}\cdot\left(\frac{\mathcal{C}}{n}\right)^n\right]}$	$\begin{cases} P\left[1+\mathcal{A}\left(\left(\frac{\mathcal{C}_2}{r_2}\right)^2-1\right)\right], & r_1\geq\mathcal{C}_1 \\ P\Big/\left[1+\mathcal{A}\left(\left(\frac{\mathcal{C}_1}{r_1}\right)^2-1\right)\right], & r_1\leq\mathcal{C}_1 \end{cases}$
Post-swap token$_1$ reserve r_1'	r_1+x_1				
Post-swap token$_2$ reserve r_2'	$\frac{\mathcal{C}}{r_1'}$	$\frac{\frac{\mathcal{C}_1\mathcal{C}_2}{(1-\frac{1}{\sqrt{\mathcal{A}}})^2}}{\left(r_1'+\frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}\right)}-\frac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1}$	$r_2\left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}$	$\frac{\sqrt{\frac{4\mathcal{C}\left(\frac{\mathcal{C}}{n}\right)^n}{\mathcal{A}\cdot\prod_{k\neq 2}'}+\left[\left(1-\frac{1}{\mathcal{A}}\right)\mathcal{C}-\sum_{k\neq 2}'\right]^2}+\left(1-\frac{1}{\mathcal{A}}\right)\mathcal{C}-\sum_{k\neq 2}'}{2}$	$\begin{cases} \frac{\mathcal{C}_1-r_1'+P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})+\sqrt{\left[\mathcal{C}_1-r_1'+P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})\right]^2+4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2}}{2P\cdot(1-\mathcal{A})}, & r_1'\geq\mathcal{C}_1 \\ \mathcal{C}_2+\frac{(\mathcal{C}_1-r_1')\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]}{P}, & r_1'\leq\mathcal{C}_1 \end{cases}$
Swap amount x_2	$r_2'-r_2$				
Slippage $S(x_i,\{r_k\};\mathcal{C})=\frac{x_i/x_o}{{}_iE_o}-1$	$\frac{x_1}{r_1}$	$\frac{x_1}{r_1+\frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}}$	$\frac{\frac{x_1}{r_1}\cdot\frac{w_1}{w_2}}{1-\left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}}-1$	$\frac{\frac{x_1\cdot\left[\mathcal{A}\cdot r_1\cdot\prod_k r_k+\mathcal{C}\cdot\left(\frac{\mathcal{C}}{n}\right)^n\right]}{r_1\cdot\left[\mathcal{A}\cdot r_2\cdot\prod_k r_k+\mathcal{C}\cdot\left(\frac{\mathcal{C}}{n}\right)^n\right]}}{1-\frac{\sqrt{\frac{4\mathcal{C}\left(\frac{\mathcal{C}}{n}\right)^n}{a\cdot\prod_{k\neq 2}'}+\left[\left(1-\frac{1}{a}\right)\mathcal{C}-\sum_{k\neq 2}'\right]^2}+\left(1-\frac{1}{a}\right)\mathcal{C}-\sum_{k\neq 2}'}{2r_2}}-1$	$\begin{cases} \frac{2\cdot(1-\mathcal{A})\cdot x_1}{r_1'-\mathcal{C}_1+\mathcal{C}_2\cdot P-\sqrt{\left[\mathcal{C}_1-r_1'+P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})\right]^2+4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2}}-1, & r_1'\geq\mathcal{C}_1 \\ \frac{x_1}{(r_1'-\mathcal{C}_1)\cdot\left[1+A\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]}-1, & r_1'\leq\mathcal{C}_1 \end{cases}$
Divergence loss $=\frac{V'(\rho,\{r_k\};\mathcal{C})}{V_{\text{held}}(\rho;\{r_k\},\mathcal{C})}-1$	$\frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}}-1$	$\begin{cases} \frac{(\rho+1)\cdot\sqrt{\mathcal{A}}-1}{2+\rho}, & -1\leq\rho\leq\frac{1}{\mathcal{A}}-1 \\ \frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}}-1 \\ \frac{1}{1-\frac{1}{\sqrt{\mathcal{A}}}}, & \frac{1}{\mathcal{A}}-1\leq\rho\leq\mathcal{A}-1 \\ \frac{\sqrt{\mathcal{A}}-1-\rho}{2+\rho}, & \rho\geq\mathcal{A}-1 \end{cases}$	$\frac{(1+\rho)^{w_2}}{1+w_2\cdot\rho}-1$	Complex	0 at equilibrium

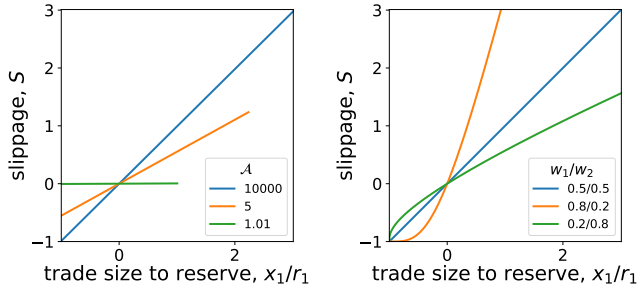


(a) Uniswap V2 & 3, Equation 19 (b) Balancer, Equation 36

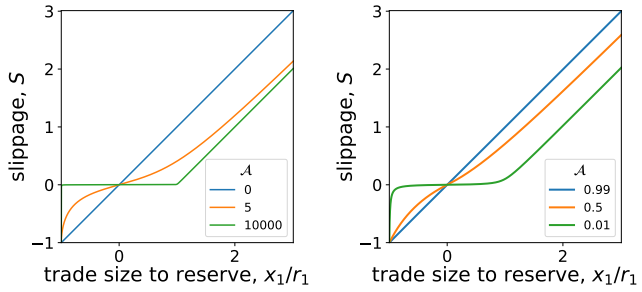


(c) Curve, Equation 44 (d) DODO, Equation 50

Figure 3: Conservation function of different AMMs



(a) Uniswap V2 & 3, Equation 22 (b) Balancer, Equation 39



(c) Curve, Equation 48 (d) DODO, Equation 53

Figure 4: Slippage function of different AMMs

B. Other AMM protocols

1) Sushiswap: Sushiswap is a fork of Uniswap, though the two mainly differ in governance token structure and user experience. The conservation function, slippage and divergence loss are identical to the Uniswap protocol.

In August 2020, Sushiswap gained a considerable portion of Uniswap's liquidity by conducting a so-called "vampire attack", where Sushiswap users were incentivized to provide Uniswap liquidity tokens into the Sushiswap protocol and rewarding them with `SUSHI` tokens [14]. After Uniswap launched `UNI` and its corresponding liquidity mining program, Uniswap is back on track with its growth schedule. Nevertheless, as shown in Table II, Sushiswap remains a popular choice for users to deposit their funds.

2) QuickSwap: Previous sections only cover AMMs on the Ethereum native blockchain, but AMM protocols also gain popularity on Ethereum sidechains. QuickSwap [15] is a Uniswap clone that went live in February 2021 on Polygon (previously Matic Network). Polygon [16] is a protocol and framework for building and connecting Ethereum-compatible blockchain networks, called a "Layer 2 aggregator" of multiple "Layer 2 solutions" such as Optimistic Rollups and zkRollups. TVL on QuickSwap has reached more than $100M in March 2021, with 24h volumes peaking at $30M decreasing to $10M at the beginning of April [17].

3) Bancor V2.1: While Bancor's white paper [18] gives the impression that a different conservation function is applied, a closer inspection of their transaction history and smart contract leads to the conclusion that Bancor is using the same formula as Balancer.[4] As the vast majority of bancor pools consist of two assets, one of which is usually `BNT`, with the reserve weights of 50%–50%, Bancor's swap mechanism is equivalent to Uniswap. Bancor V2.1 now allows single-sided asset exposure, and provides divergence loss insurance [19] (see IV-B4c).

4) Additional AMM features:

a) Time component: A time component refers to the ability to change traditionally fixed hyperparameters over time. Balancer V1 and V2 implement this (Table IV), by allowing liquidity pool creators to set a scheme that changes the weights of two pool assets over time. This implementation is called a Liquidity Bootstrapping Pool and is discussed in IV-C.

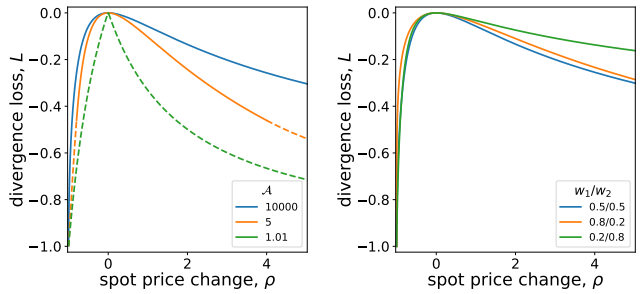
b) Dynamic swap fee: Dynamic fees are introduced by Kyber 3.0 to reduce the impact of divergence loss for LPs. The idea is to increase swap fees in high-volume markets and reduce them in low-volume markets. This should result in more protection against divergence loss, as during periods of sharp token price movements during a high-volume market, LPs absorb more fees. In low-volume and low-volatility markets, trading is encouraged by lowering the fees.

c) Divergence loss insurance: Popularized by Bancor V2.1, LPs are insured against impermanent loss after 100 days in the pool, with a 30-day cliff at the beginning. Bancor achieves this by using an elastic `BNT` supply that allows the protocol to co-invest in pools and pay for the cost of impermanent loss with swap fees from its co-investments [20]. This insurance policy is earned over time, 1% each day that liquidity is staked in the pool.

[4]This has been confirmed by a developer in the Bancor Discord community

Table IV: Overview of important existing AMM protocols on Ethereum, Solana, Polkadot, Tezos and EOS. CS = Constant-Sum, CP = Constant-Product, OP = Oracle price component, CC = Capital concentration, TD = Time component.

		Conservation function			AMM add-ons			Associated Attacks			
Protocol	**Pool structure**	**CP**	**CS**	**OP**	**CC**	**T**	**Divergence loss compensation**	**Flash loan attack**	**Vampire attack**	**Chain**	**Mainnet launch**
Uniswap V1 [21]	asset-pair	●	○	○	○	○	—	[22]	—	Ethereum	11/2018
Uniswap V2 [23]	asset-pair	●	○	○	○	○	—	[24], [25], [26], [27]	—	Ethereum	05/2020
Uniswap V3 [28]	asset-pair	●	○	○	●	○	—	—	—	Ethereum	05/2021
Balancer V1 [3]	multi-asset	●	○	○	○	●	—	[29]	—	Ethereum	03/2020
Balancer V2 [30]	multi-asset	●	○	○	○	●	—	—	—	Ethereum	—
Curve [10]	multi-asset	●	●	○	○	○	—	[24], [26]	—	Ethereum	01/2020
DODO [7]	single-asset	●	○	●	○	○	—	—	—	Ethereum, BSC	09/2020
Bancor V1 [18]	asset-pair	●	○	○	○	○	—	—	—	Ethereum, EOS	06/2017
Bancor V2 [19]	asset-pair	●	○	●	○	○	—	—	—	Ethereum, EOS	04/2020
Bancor V2.1 [4]	asset-pair	●	○	○	○	○	Divergence loss insurance	—	—	Ethereum, EOS	10/2020
SushiSwap [14]	asset-pair	●	○	○	○	○	—	[26], [27]	[31]	Ethereum	08/2020
Mooniswap [32]	asset-pair	●	○	○	○	●	—	—	—	Ethereum	08/2020
mStable [8]	asset-pair	○	●	○	○	○	—	—	—	Ethereum	07/2020
Kyber 3.0 [33]	multi-asset	●	○	○	●	○	Dynamic swap fee	—	—	Tezos	—
StableSwap [34]	multi-asset	●	●	○	○	○	—	—	—	Solana	—
TrueSwap [35]	asset-pair	●	○	○	○	○	—	—	—	Tron	—
HydraDX [36]	multi-asset	●	○	●	●	○	—	—	—	Polkadot	—



(a) Uniswap V2 & 3, Equation 26 (b) Balancer, Equation 43

(c) Curve, Equation 18 (d) DODO, no div. loss

Figure 5: Divergence loss of different AMMs

C. DeFi protocols with AMM implementations

AMMs form the basis of other protocols that implement existing or invent newly designed bonding curves, facilitating the functionalities of these implementing protocols. In this section, we present few examples of projects that use AMM designs under the hood.

1) Gyroscope Protocol: Gyroscope Protocol [37] is a stablecoin backed by a reserve portfolio that diversifies all DeFi risks. Gyro Dollars can be minted for a price near $1 and can be redeemed for an amount worth of near $1 in reserve assets, as determined through a new Automated Market Maker (AMM) design that balances risk in the system.

Gyroscope includes a Primary-market AMM (P-AMM), through which Gyro Dollars are minted and redeemed, and a Secondary-market AMM (S-AMM) for Gyro Dollar trading. Similar to Uniswap V3, where a price range constraint is imposed. The P-AMM yields a mint quote and a redeem quote that serves as a price range constraint for the S-AMM to decide upon concentrated liquidity ranges [38].

2) EulerBeats: EulerBeats [39] is a protocol that issues limited edition sets of algorithmically generated art and music, based on the Euler number and Euler totient function. The project uses self-designed bonding curves to calculate burn prices of music/art prints, depending on the existing supply. The project thus implements a form of AMM to mint and burn NFTs price-efficiently.

3) Pods Finance: Pods [40] is a decentralized non-custodial options protocol that allows users to create calls and or puts and trade them in the Options AMM. Users can participate as sellers and buy puts and calls in a liquidity pool or act as liquidity providers in such a pool. The specific AMM is one-sided and built to facilitate an initially illiquid options market and price option algorithmically using the Black-Scholes pricing model. Users can effectively earn fees by providing liquidity, even if the options are out-of-the-money, reducing the cost of hedging with options.

4) Balancer Liquidity Bootstrapping Pool: Liquidity Bootstrapping Pools (LBPs) are pools where controllers can change the parameters of the pool in controlled ways, unlike immutable pools described in section Section IV. The idea of an LBP is to launch a token fairly, by setting up a two-token

pool with a project token and a collateral token. The weights are initially set heavily in favour of the project token, then gradually "flip" to favour the collateral coin by the end of the sale. The sale can be calibrated to keep the price more or less steady (maximizing revenue) or declining to the desired minimum (e.g., the initial offering price) [41].

5) YieldSpace: The YieldSpace paper [42] introduces an automated liquidity provider for fixed yield tokens. A formula called the "constant power sum invariant" incorporates time to maturity as input and ensures that the liquidity provider offers a constant interest rate—rather than price—for a given ratio of its reserves. `fyTokens` are synthetic tokens that are redeemable for a target asset after a fixed maturity date [43]. The price of a `fyToken` floats freely before maturity, and that price implies a particular interest rate for borrowing or lending that asset until the `fyToken` 's maturity. Standard AMM protocols as discussed in Section IV are capital-inefficient. By introducing the concept of a constant power sum formula, the writers want to build a liquidity provision formula that works in "yield space" instead of "price space".

6) Notional Finance: Notional Finance [44] is a protocol that facilitates fixed-rate, fixed-term crypto-asset lending and borrowing. Fixed interest rates provide certainty and minimize risk for market participants, making this an attractive protocol among volatile asset prices and yields in DeFi. Each liquidity pool in Notional refers to a maturity, holding `fCash` tokens attached to that date. For example, `fDai` tokens represent a fixed amount of `DAI` at a specific future date. The shape of the Notional AMM follows a logit curve, to prevent high slippage in normal trading conditions. Three variables parameterize the AMM: the scalar, the anchor, and the liquidity fee [45]. The first and second mentioned allowing for variation in the steepness of the curve and its position in a xy-plane, respectively. By converting the scalar and liquidity fee to a function of time to maturity, fees are not increasingly punitive when approaching maturity.

7) Gnosis Custom Market Maker: The Gnosis CMM [46] allows users to set multiple limit orders at custom price brackets and passively provide liquidity on the Gnosis Protocol. The mechanism used is similar to the Uniswap V3 structure, although it allows for even more possibilities to market makers by allowing them to choose price upper and lower limits and a number of brackets within that price range. Uniswap V3 allows liquidity providers to solely choose the upper and lower limits. Because users deposit funds to the assets at different price levels specifically, this specific application behaves more like a central limit order book than an AMM pool.

D. Discussion

If one needs to trade similarly priced assets, then Curve does that the best. Suppose it concerns an ETF-like portfolio, which automatically re-balances, Balancer will help. As mentioned before, Balancer has the same slippage and divergence loss formulas as Uniswap in case of an equal 0.5/0.5 split, while Curve has almost identical formulas in case of $\mathcal{A} \to 0$, as can be seen in Figure 5. On Balancer, when the user deals with strongly unequally divided pool weights, the trade size has a relatively high impact on the slippage compared to a pool with equal weights, as is shown in Figure 4b. Conversely, divergence loss is less impactful in that situation since arbitrageurs are eating fewer profits from the liquidity providers. In Curve, the bigger $\mathcal{A}$ is, the smaller the price slippage should be, but the more significant the divergence loss is in case of spot price changes, as shown in Figure 4c and Figure 5c. It must be noted that because all assets in a Curve pool are backed by the same asset, spot price changes should have a low impact, since all assets in the pool are facing that same spot price change. This is why Curve has an inherent advantage when trading similar assets. From a theoretical point of view, it seems like DODO can offer similar functionalities as Curve, as seen in Figure 3 and Figure 4, but without having the disadvantage of divergence loss for liquidity providers. The one-to-one comparison of these protocols may not make complete sense in some cases, such as when one compares Curve and Uniswap, but it sheds light, nevertheless, on how much better it is to trade stablecoins and similarly priced assets on Curve.

Attack 1 Flash-loan-funded price oracle attack

1: **Take a flash loan** to borrow x_A token$_A$ from a PLF, whose value is equivalent to x_B token$_B$ at market price.
2: **Swap** x_A token$_A$ for $x_B - \Delta_1$ token$_B$ on an AMM, pushing the new price of token$_A$ in terms of token$_B$ down to $\frac{x_B - \Delta_2}{x_A}$, where $\Delta_2 > \Delta_1 > 0$ due to slippage.
3: **Borrow** $x_A + \Delta_3$ token$_A$ with $x_B - \Delta_1$ token$_B$ as collateral on a PLF that uses the AMM as their sole price oracle. To temporarily satisfy overcollateralization, $\frac{x_B - \Delta_2}{x_A} < \frac{x_B - \Delta_1}{x_A + \Delta_3}$.
4: **Repay the flash loan** with x_A token$_A$.

V. Security and privacy issues with AMM

A. AMM-associated attacks

1) Oracle attack: At the end of 2020, a series of flash loan-funded price oracle attacks caused exploits in numerous protocols. In this kind of attack, adversaries manipulate protocols that use a DEX as their sole price oracle.

Following Attack algorithm 1, an attacker profits with Δ_3 token$_A$ less any transaction fees incurred. The attack temporarily distorts the price of token$_A$ relative to token$_B$. After the prices are arbitraged back, the attack would leave the loan taken from step 3 undercollateralized, jeopardizing the safety of lenders' funds on PLF. Examples of such attacks are exploits on Harvest finance [24], Value DeFi [26] and Cheese bank [25].

This broken design can generally be fixed by either providing time-weighted price feeds, or using external decentralized oracles. The first solution ensures that a price feed cannot be manipulated within the same block, while the second solution aggregates price data from multiple independent data providers that add a layer of security behind the aggregation algorithm, makes sure that prices are not easily manipulated.

2) Rug pull: The general idea of a rug pull is to lure people into buying the coin with no value, subsequently swapping this coin for ETH or another cryptocurrency with value, as shown in Attack algorithm 2. One method is to create a coin with the same name as an existing one. This attracts a lot of attention since everyone wants to pick up the coin at the lowest price possible. The coin is being bought up, and the original liquidity provider swaps his fake coin for ETH. There are other ways rug pulls are performed. One of the co-authors of this paper was used as a marketing pawn in one ploy. The creators of the fake token send it out to several prominent people, creating false hype. Potential buyers see that major buyers have purchased the token and start buying themselves. They very quickly realize that the token cannot be swapped back for ether. Sometimes, the attackers let people trade the coin back for ether, but only for a short period since they are running the risk of losing money. One example of this is the **RAM** token (address: `0x90b7a437ddaf1d5686445b928da82d86dd447ec5`). The attacker extracted 24 ETH from the rug pull.

Attack 2 Rug Pull

1: **Mint** a new coin XYZ.
2: **Create** a liquidity pool with $x_{\texttt{XYZ}}$ XYZ and $x_{\texttt{ETH}}$ ETH (or any other valuable cryptocurrency) on an AMM, and receive LP tokens.
3: **Attract** unwitting traders to buy XYZ with ETH from the pool, effectively changing the composition of the pool.
4: **Withdraw** liquidity from the pool by surrendering LP tokens, and obtain $x_{\texttt{XYZ}} - \Delta_1$ XYZ and $x_{\texttt{ETH}} + \Delta_2$ ETH, where $\Delta_1, \Delta_2 > 0$.

3) Frontrunning: Frontrunners place their trade immediately before someone else's trade. These are usually the traders that attempt to get the best price of a new coin before anyone else. They then sell these coins onto the market. Almost all Polkastarter IDOs are frontran on Uniswap. Each listing brings the attacker at least $1 million in profits. Sometimes, the attackers use Flashbots[5] for this. Most of the time, they spam the block in a fashion similar to how backrunners fill the block. This is done to definitively achieve nonce index that immediately follows the nonce of the transaction that unpauses trading on Uniswap. The attacker buys up almost all of the token supply. Since there is hype, this does not stop retailers from buying at exorbitant prices. However, now the seller with the most significant supply is the attacker, and he swaps the purchased coin for ether supplied by the retail. These attacks are incredibly vicious since they motivate more Polkastarter IDOs. In a sense, this is value extraction from the Ethereum blockchain.

A simple way to identify these attacks is to observe new listings on Uniswap and observe the transaction immediately following the unpause transaction in the ERC20 coin. You will find that the attacker will be on most of those buying at least 200-300 ETH worth of the just listed token.

Normal exchange users could set a low slippage tolerance to avoid suffering from a price elevated by front-runners. However, an overly low slippage tolerance may lead a transaction to fail, resulting in a waste of gas fee.

4) Backrunning: Backrunners place their trade immediately after someone else's trade. The attacker needs to fill up the block with a large number of cheap gas transactions to definitively follow the target's transaction. In comparison, frontrunning requires a single high valued transaction. Frontrunning is detrimental to the user, in contrast, backrunning is detrimental to the network as a whole and so has more negative externalities.[6]

There are a number of agents in this flow. There needs to be a miner and / or the "extractor". One way to extract the value is for the miner to amend the order of the transactions and place his. For example, a big trade on Uniswap with high slippage will be sniped by placing the transactions around it. See detailed example below.

Attack 3 Sandwich price attack

1: User$_A$ wishes to purchase x_A XYZ whose spot price is P_1 on an AMM with gas fee g_1.
2: User$_B$ **observes** the mempool and sees the transaction
3: User$_B$ **front-runs** by buying x_B XYZ with a higher gas fee $g_2 > g_1$ on the same AMM .
4: User$_B$ and User$_A$'s transactions are executed sequentially at respective average price of P_B and P_A, pushing XYZ's spot price up to P_2, where $P_2 > P_A > P_B > P_1$ due to slippage.
5: User$_B$ **back-runs** by selling x_B XYZ at an average price of P'_B, with $P_2 > P'_B > P_B$ due to slippage.

5) Sandwich attacks: Combing front- and back-running, an adversary of a sandwich attack places his orders immediately before and after the victim's trade transaction. The attacker uses front-running to cause victim losses, and then uses back-running to pocket benefits. Zhou et al. [1] detail two sandwich attacks that can occur on an AMM: one exchange user attacking another, and an LP attacking an exchange user.

Attack algorithms 3 and 4 describe those two attacks.

6) Vampire attack: As mentioned in IV-B1, Sushiwap launched in August 2020 as a fork of Uniswap and gained a lot of traction by allowing users to deposit their Uniswap LP tokens in Sushiswap in return for rewards, thereby siphoning out liquidity from Uniswap, a sequence later called a "Vampire Attack" [47].

In a first step of a Vampire attack, a new protocol B incentives liquidity providers of another platform A to stake their LP tokens into protocol B. In case of Sushiswap, Uniswap LPs were rewarded with SUSHI tokens when they staked their LP tokens into the Sushiswap protocol. In the second stage, a migration of liquidity happens from protocol A to protocol

[5]https://github.com/flashbots/pm

[6]https://github.com/ethereum/go-ethereum/issues/21350

Attack 4 Sandwich LP attack

1: User$_A$ wishes to purchase x_A token$_A$ with token$_B$ using an AMM pool of r_A token$_A$ and r_B token$_B$ with gas fee g_1.
2: LP$_B$ **observes** the mempool and sees the transaction.
3: LP$_B$ **front-runs** by withdrawing liquidity $k\,r_A$ token$_A$ and $k\,r_B$ token$_B$ with a higher gas fee $g_2 > g_1$.
4: LP$_B$ and User$_A$'s transactions are executed sequentially, resulting in a new composition of the pool with $(1-k)r_A + x_A$ token$_A$ and $(1-k)r_B - x_B$ token$_B$.
5: LP$_B$ **back-runs** by re-providing $k\,r_A$ token$_A$ and $k \cdot \frac{(1-k)r_B - x_B}{(1-k)r_A + x_A}$ token$_B$.
6: LP$_B$ **back-runs** by selling $(1 - \frac{(1-k)r_B - x_B}{(1-k)r_A + x_A})$ token$_B$ for some token$_A$

B. By doing this, protocol B now has sucked liquidity from protocol A and moved it to its own contracts, giving access to more volume and thus creating a more attractive proposition to users. The migration process was executed by a smart contract that took the Uniswap LP tokens in Sushiswap, converted those to the represented liquidity on Uniswap, transferred the assets to Sushiswap and got Sushiswap LP tokens in return. Effectively, Sushiswap migrated liquidity from Uniswap to Sushiswap on 9 September 2020 [31], thereby moving $830 million to the new born AMM. Users' Uniswap LP tokens got automatically replaced with Sushiswap LP tokens, representing the same liquidity. To encourage liquidity providers to participate in the migration, Sushi continued to reward LPs and users that were staking `SUSHI`.

B. Privacy concerns

On AMM DEX, security problems often go hand in hand with privacy issues. Among the named attacks from the previous section, the sandwich attacks are enabled by the transparency and openness of public blockchains such as Ethereum, where transactions are observable to everyone. Against this backdrop, on-chain privacy-preserving services and products are on the rise. For example, Blank [48], a non-custodial Ethereum browser extension wallet, offers IP protection and transaction obfuscation; Enigma [49] builds a network of "secret nodes" that can perform computations on encrypted data without the necessity to expose original raw data.

C. Public information on attacks

We have found that the current MEV dashboard[7] includes but a tiny subset of maximum extractable value. Only single transaction externalities are logged. Our experience showed that initial coin listings are pulling at least the daily dashboard quoted MEV alone. An exciting avenue would be to explore this area more closely since that would mean highly optimistic reported DEX volumes.

[7]https://explore.flashbots.net

VI. Related work

A. Blockchain-based DEXs

Our work is first and foremost related to the literature body covering blockchain-based DEXs.

1) Security: Qin et al. [50] conduct empirical analyses on various AMM attacks, including transaction (re)ordering and front-running, and demonstrate the profitability in performing transaction replay through a simple trading bot. Security risk in terms of attack vectors in high-frequency trading on DEXs are discussed in Zhou et al. [1], and Qin et al. [51]. Flash loan attacks with the aid of AMMs on Ethereum are described in Cao et al. [52], Perez et al. [53] and Wang et al. [54]. Victor et al. [55] detect self-trading and wash trading activities on order-book based DEXs. Gudgeon et al. [56] explore design weaknesses and volatility risks in AMM DEXs.

2) Privacy: Angeris et al. [57] argue that privacy is impossible with typical constant-function market makers and propose several mitigating strategies. Stone et al. [58] describe a protocol that allows trustless, privacy-preserving cross-chain cryptocurrency transfers but is yet susceptible to vampire attacks.

3) Protocol mechanism: Angeris et al. [59] discuss arbitrage behaviour and price stability in constant product and constant mean markets. Lo et al. [60] empirically evidence that the simplicity of Uniswap ensures the ratio of reserves to match the trading pair price. Despite historical oracle attacks associated with AMMs (see Section V), Angeris et al. [59], [61] show that constant-function-AMM users are incentivized to correctly report the price of an asset, suggesting the suitability for those AMMs to act as a decentralized price oracle for other DeFi protocols. Angeris et al. [13] present a method for constructing constant-function AMM whose portfolio value functions match an arbitrary payoff.

B. DEX and AMM in the context of market microstructure

As two core topics of market microstructure [62], decentralized exchange and market-making have been intensively covered in the discipline of financial economics long before the emergence of blockchain.

1) DEX: Existing literature primarily suggests the higher efficiency of DEX markets over centralized ones.

Perraudin et al. [63] investigate decentralized forex markets and conclude that DEXs are efficient when different market makers can transact with each other and that decentralized markets are more immune to crashes than centralized ones. Nava [64] analyzes quantity competition in the decentralized oligopolistic market and suggest perfect competition can be approximated in large rather than small DEX markets. Malamud et al. [65] develops an equilibrium model of general DEX and prove that decentralized markets can more efficiently allocate risks to traders with heterogeneous risk appetites than centralized ones.

2) AMM: The concept of automated market making can be traced back to Hanson's logarithmic market scoring rules (LMSR) [66], [67]. LMSR has since been refined and compared to alternative market-making strategies.

Othman et al. [68] address non-sensibility to liquidity and non-profitability of LMSR market making. They propose a bounded, liquidity-sensitive AMM that runs with a profit by levying transaction cost to subsidize liquidity, a strategy later widely implemented by blockchain-based DEXs with AMM protocols to compensate for divergence loss (see II-C3b) experienced by LPs. Brahma et al. [69] propose a Bayesian market maker for binary markets which exhibit better convergent behaviour at equilibrium than LMSR.

Jumadinova et al. [70] compare LMSR with different AMM strategies, including myopically optimizing market-maker, reinforcement learning market maker and utility-maximizing market maker. Simulating empirical market data, they find that reinforcement learning-based AMM outperforms other strategies in terms of maintaining low spread while simultaneously obtaining high utilities. Slamka [71] compare LMSR with dynamic parimutuel market (DPM), dynamic price adjustments (DPA) and an AMM by the Hollywood stock exchange (HSX) in the context of prediction markets. They show that LMSR and DPA generate the highest forecast accuracy and lowest losses for market operators. Today, LMSR has become the de facto AMM for prediction markets [72] and was adopted by the Ethereum-based betting platform Augur [73].

Wang [72] compares mathematical models for AMMs, including LMSR, liquidity sensitive LMSR (LS-LMSR) and common constant-function AMMs, and proposes constant circle/ellipse based cost functions for superior computational efficiency. Capponi et al. [74] analyze the market microstructure of constant-product AMMs, and predict that AMMs will be used more for low-volatility tokens.

VII. Conclusion

The DeFi ecosystem is a relatively new concept, and innovations within the space are being developed at an incredible speed. AMMs are an incredible innovation sprung up by the trustless, verifiable and censorship-resistant decentralized Turing complete and global execution machine. As the use of Decentralized Exchanges becomes increasingly crucial in the industry of Decentralized Finance, it is essential to understand how different flavours of those protocols may fit in a general AMM framework and how the significant players differentiate themselves given that framework.

We have systematized the knowledge around Automated Market Makers during this research and applied that expertise to several exchanges: Uniswap, Balancer, Curve and DODO and made comments on other AMM protocols: Sushiswap, QuickSwap, Bancor V2.1 and others. We use state-space representation to formalize and generalize the AMM algorithms. Existing protocols can be explored more in-depth using this framework. Our research summarizes the economics and risks in AMMs. We find that non-order book DEXs are highly susceptible to a plethora of economic risks such as wash trading, frontrunning, backrunning, sandwiching, rug pulling.

Future research into AMM mechanisms can build upon this systematization of knowledge and establish unique ways for differentiating and sustaining AMM innovations.

Appendix

A. Protocol formulas

1) Uniswap V2:

a) Conservation function: The product of reserve quantity of token$_1$, r_1, and reserve quantity of token$_2$, r_2, stays constant with swapping:

$$\mathcal{C} = r_1 \cdot r_2 \tag{19}$$

b) Spot exchange rate: Given the *equal value assumption* encoded in the pool smart contract, the implied spot price of assets in a liquidity pool can be derived based on the ratio between their reserve quantities. Specifically, denominated in token 1, the price of token$_2$ can be expressed as:

$$_1E_2 = \frac{r_1}{r_2} \tag{20}$$

c) Swap amount: Based on the Uniswap conservation function (Equation 19), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$) can be calculated following the steps described in III-C3:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \frac{\mathcal{C}}{r_1'} \\ x_2 &= r_2 - r_2' \end{aligned} \tag{21}$$

d) Slippage: The slippage that a Uniswap user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \frac{x_1/x_2}{_1E_2} - 1 = \frac{x_1}{r_1} \tag{22}$$

Figure 4a illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

e) Divergence loss: Given the equal value assumption with Uniswap, the reserve value of token 1, V_1, equals exactly half of original value of the entire pool V (token$_1$ being numéraire):

$$\frac{V}{2} = V_1 = V_2 = r_1 \tag{23}$$

Should a liquidity provider have held r_1 token$_1$ and r_2 token$_2$, then when token$_2$ appreciates by ρ (depreciates when $\rho < 0$), the total value of the original reserve composition V_{held} becomes:

$$V_{\text{held}} = V + V_2 \cdot \rho = r_1 \cdot (2 + \rho) \tag{24}$$

With r_1 token$_1$ and r_2 token$_2$ locked in a liquidity pool from the beginning, their quantity ratio would have been updated through users' swapping to result in token$_2$'s price change of ρ. The equal value assumption still holds, and the updated pool value V' becomes:

$$\frac{V'}{2} = V_1' = V_2' = r_1' = r_1 \cdot \sqrt{1+\rho} \tag{25}$$

Note that $r_2' = \frac{r_2}{\sqrt{1+\rho}}$ and $p' = \frac{(1+\rho)r_1}{r_2}$, which preserves the invariance of $\mathcal{C}$, and reflects the change in token$_2$'s spot exchange rate against token$_1$.

As illustrated in Figure 3a, the divergence loss due to liquidity provision as opposed of holding can thus be expressed as a function of price change:

$$L(\rho) = \frac{V'}{V_{\text{held}}} - 1 = \frac{\sqrt{1+\rho}}{1+\frac{\rho}{2}} - 1 \tag{26}$$

2) Uniswap V3:

a) Conservation function: Suppose a liquidity provider supplies $\mathcal{C}_1$ token$_1$ and $\mathcal{C}_2$ token$_2$, with the restriction that his liquidity is only used for a specific range of exchange rates: $[\frac{\mathcal{C}_1}{\mathcal{C}_2 \cdot \mathcal{A}}, \frac{\mathcal{C}_1 \cdot \mathcal{A}}{\mathcal{C}_2}]$ where $\mathcal{A} > 1$ and the initial exchange rate equals $\frac{\mathcal{C}_1}{\mathcal{C}_2}$. The shape of the conservation function is then *identical* to liquidity provision of the following amounts under Uniswap V2:

$$r_1^{\text{equiv}} = \frac{\mathcal{C}_1}{1 - \frac{1}{\sqrt{\mathcal{A}}}} \quad \text{and} \quad r_2^{\text{equiv}} = \frac{\mathcal{C}_2}{1 - \frac{1}{\sqrt{\mathcal{A}}}}$$

The bonding curve of a Uniswap V3 pool is equivalent to that of a Uniswap V2 one moving left along the x-axis by $(r_1^{\text{equiv}} - \mathcal{C}_1)$ and down along the y-axis by $(r_2^{\text{equiv}} - \mathcal{C}_2)$. Thus, Uniswap V3 conservation function can be expressed as:

$$\begin{aligned} [r_1 + (r_1^{\text{equiv}} - \mathcal{C}_1)] \cdot [r_2 + (r_2^{\text{equiv}} - \mathcal{C}_2)] &= r_1^{\text{equiv}} \cdot r_2^{\text{equiv}} \\ \left(r_1 + \frac{\mathcal{C}_1}{\sqrt{\mathcal{A}} - 1}\right) \cdot \left(r_2 + \frac{\mathcal{C}_2}{\sqrt{\mathcal{A}} - 1}\right) &= \frac{\mathcal{A} \cdot \mathcal{C}_1 \cdot \mathcal{C}_2}{(\sqrt{\mathcal{A}} - 1)^2} \end{aligned} \tag{27}$$

where $0 \le r_1 \le \mathcal{C}_1 \cdot (\sqrt{A} + 1)$ and $0 \le r_2 \le \mathcal{C}_2 \cdot (\sqrt{A} + 1)$.

b) Exchange rate:

$$_1E_2 = \frac{r_1 + \frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}}{r_2 + \frac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1}} \tag{28}$$

Note that when token$_1$ is depleted, i.e. $r_1 = 0$, then

$$\begin{aligned} r_2 + \tfrac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1} &= \tfrac{\mathcal{A} \cdot \mathcal{C}_2}{\sqrt{\mathcal{A}}-1} \\ _1E_2 &= \tfrac{\mathcal{C}_1}{\mathcal{C}_2 \cdot \mathcal{A}} \end{aligned}$$

Similarly, when token$_2$ is depleted, i.e. $r_2 = 0$, then $_1E_2 = \frac{\mathcal{C}_1}{\mathcal{C}_2 \cdot \mathcal{A}}$

c) Swap amount: The swap amount can be derived from the conservation function Equation 27:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \tfrac{\mathcal{C}_1 \mathcal{C}_2}{(1 - \frac{1}{\sqrt{\mathcal{A}}})^2} \Big/ \left(r_1' + \tfrac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}\right) - \tfrac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1} \\ x_2 &= r_2 - r_2' \end{aligned} \tag{29}$$

d) Slippage: The slippage should have the same magnitude as in Uniswap V2, but with r_1 amplified by an increase of $\frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}$:

$$S(x_1) = \frac{x_1/x_2}{_1E_2} - 1 = \frac{x_1}{r_1 + \frac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1}} \tag{30}$$

e) Divergence loss: Using the intermediary results from A1e, we can easily derive V_{held}, r_1' and r_2', and subsequently V':

$$V_{\text{held}} = C_1 \cdot (2 + \rho) \tag{31}$$

$$\begin{aligned} r_1' &= r_1^{\text{equiv}} \sqrt{1+\rho} - \tfrac{\mathcal{C}_1}{\sqrt{\mathcal{A}}-1} = \tfrac{\mathcal{C}_1 \cdot (\sqrt{1+\rho} - \frac{1}{\sqrt{\mathcal{A}}})}{1 - \frac{1}{\sqrt{\mathcal{A}}}} \\ r_2' &= \tfrac{r_2^{\text{equiv}}}{\sqrt{1+\rho}} - \tfrac{\mathcal{C}_2}{\sqrt{\mathcal{A}}-1} = \tfrac{\mathcal{C}_2 (\frac{1}{\sqrt{1+\rho}} - \frac{1}{\sqrt{\mathcal{A}}})}{1 - \frac{1}{\sqrt{\mathcal{A}}}} \\ V' &= V_1' + V_2' = r_1' + \tfrac{\mathcal{C}_1 (1+\rho) r_2'}{\mathcal{C}_2} = \tfrac{\mathcal{C}_1 (2\sqrt{1+\rho} - \frac{2+\rho}{\sqrt{\mathcal{A}}})}{1 - \frac{1}{\sqrt{\mathcal{A}}}} \end{aligned} \tag{32}$$

When $-1 \le \rho \le \frac{1}{\mathcal{A}} - 1$, then token$_1$ becomes depleted, and the liquidity provider is left with token$_2$:

$$V' = \frac{\mathcal{C}_1 (1+\rho)}{\mathcal{C}_2} \cdot r_2' = \mathcal{C}_1 \cdot (1+\rho) \cdot \left(\sqrt{\mathcal{A}} + 1\right) \tag{33}$$

When $\rho \ge \mathcal{A} - 1$, then token$_2$ becomes depleted, and the liquidity provider is left with token$_1$:

$$V' = r_1' = \mathcal{C}_1 \cdot \left(\sqrt{\mathcal{A}} + 1\right) \tag{34}$$

The divergence loss can thus be calculated as:

$$\begin{aligned} L(\rho) &= \frac{V'}{V_{\text{held}}} - 1 \\ &= \begin{cases} \frac{(\rho+1) \cdot \sqrt{\mathcal{A}} - 1}{2+\rho}, & -1 \le \rho \le \frac{1}{\mathcal{A}} - 1 \\ \frac{\sqrt{1+\rho}}{\frac{1+\frac{\rho}{2}}{1 - \frac{1}{\sqrt{\mathcal{A}}}}} - 1, & \frac{1}{\mathcal{A}} - 1 \le \rho \le \mathcal{A} - 1 \\ \frac{\sqrt{\mathcal{A}} - 1 - \rho}{2+\rho}, & \rho \ge \mathcal{A} - 1 \end{cases} \end{aligned} \tag{35}$$

Note that $\lim_{\mathcal{A} \to 1} L(\mathcal{A} - 1) = 0$.

3) Balancer:

a) Conservation function: Balancer implements a conservation function with a weighted-product invariant (Figure 3b). Specifically, the product of reserve quantities each raised to the power of its weight stays constant with swapping:

$$\mathcal{C} = \prod_k r_k^{w_k} \tag{36}$$

b) Spot exchange rate: Given the quantity ratio $r_1 : r_2$ between token$_1$ and 2 and the implicit assumption on their value ratio $w_1 : w_2$, the price of token$_2$ denominated by token$_1$ can be expressed as:

$$_1E_2 = \frac{r_1 \cdot w_2}{r_2 \cdot w_1} \tag{37}$$

c) Swap amount: We investigate the case when a user swaps token$_1$ for token$_2$, while the reserves of all other assets remain untouched in the pool. Based on the Balancer conservation function (Equation 36), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$) can be calculated following the steps described in III-C3:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= r_2 \left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}} \\ x_2 &= r_2 - r_2' \end{aligned} \tag{38}$$

d) Slippage: The slippage that a Balancer user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$S(x_1) = \frac{x_1/x_2}{{}_1E_2} - 1 = \frac{\frac{x_1}{r_1} \cdot \frac{w_1}{w_2}}{1 - \left(\frac{r_1}{r_1'}\right)^{\frac{w_1}{w_2}}} - 1 \tag{39}$$

Figure 4b illustrates the relationship between Uniswap slippage and normalized token$_1$ reserve change $\frac{x_1}{r_1}$.

e) Divergence loss: Given the constant value ratio assumption with Balancer, the value of the entire pool V can be expressed by the reserve quantity of token 1, r_1 divided by its weight w_1 (token$_1$ being numéraire):

$$V = \frac{V_1}{w_1} = \frac{V_2}{w_2} = \frac{V_k}{w_k} = \frac{r_1}{w_1} \tag{40}$$

If token$_2$ appreciates by ρ (depreciates when $\rho < 0$) while all other tokens' prices remain unchanged, the total value of the original reserve composition, when held outside of the pool, V_{held} becomes:

$$V_{\text{held}} = V + V_2 \cdot \rho = V \cdot (1 + w_2 \cdot \rho) \tag{41}$$

With r_1 token$_1$ and r_2 token$_2$ locked in a liquidity pool from the beginning, their quantity ratio would have been updated through users' swapping to result in token$_2$'s price change of ρ. The value ratio between the pool, token$_1$ and token$_2$, remains $1 : w_1 : w_2$, and the updated pool value V' becomes:

$$V' = \frac{V_1'}{w_1} = \frac{r_1'}{w_1} = \frac{r_1 \cdot (1+\rho)^{w_2}}{w_1} = V \cdot (1+\rho)^{w_2} \tag{42}$$

The exchange rate range corresponds the liquidity provider's range requirement. Specifically, when $r_2' = \frac{r_2}{(1+\rho)^{1-w_2}}$ and $r_k' = r_k \cdot (1+\rho)^{w_2}$ for $k \neq 2$, reflecting the assumed scenario that only the value of token$_2$ appreciates by ρ, while the value of all other tokens against token$_1$ remains unchanged.

As illustrated in Figure 5b, the divergence loss due to liquidity provision as opposed to holding can thus be expressed as a function of price change:

$$L(\rho) = \frac{V'}{V_{\text{held}}} - 1 = \frac{(1+\rho)^{w_2}}{1 + w_2 \cdot \rho} - 1 \tag{43}$$

4) Curve:

a) Conservation function: As assets from the same pool are connected to the same peg by design, the ideal exchange rate between them should always equal 1. Theoretically, this could be achieved by a constant-sum invariant. Nevertheless, Curve seeks to allow an exchange rate to deviate from 1, in order to reflect the supply-demand dynamic, while simultaneously keeping the slippage low.

Curve achieves this by interpolating between two invariants, constant sum and constant product [10], with hyperparameter $\mathcal{A}$ as the interpolating factor (Equation 44).[8] When $\mathcal{A} \to 0$, the conservation function boils down to a constant-product one, as with Uniswap; when $\mathcal{A} \to +\infty$, the conservation function is essentially a constant-sum one with constant exchange rate equal to 1 (Figure 3c).

$$\mathcal{A}\left(\frac{\sum_k r_k}{\mathcal{C}} - 1\right) = \frac{\left(\frac{\mathcal{C}}{n}\right)^n}{\prod_k r_k} - 1 \tag{44}$$

b) Spot exchange rate: Rearrange Equation 44 and let

$$Z(r_1, r_2) = \frac{\left(\frac{\mathcal{C}}{n}\right)^n}{r_1 r_2 \prod_{k \neq 1,2} r_k} - 1 - \mathcal{A}\left(\frac{r_1 + r_2 + \sum_{k \neq 1,2} r_k}{\mathcal{C}} - 1\right)$$

Following III-C, the spot exchange rate can be calculated as:

$${}_1E_2 = \frac{\partial Z(r_1,r_2)/\partial r_2}{\partial Z(r_1,r_2)/\partial r_1} = \frac{r_1 \cdot \left[\mathcal{A} \cdot r_2 \cdot \prod_k r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]}{r_2 \cdot \left[\mathcal{A} \cdot r_1 \cdot \prod_k r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]} \tag{45}$$

c) Swap amount: We investigate the case when a user swaps token$_1$ for token$_2$, while the reserves of all other assets remain untouched in the pool. Based on the Curve conservation function (Equation 44), the amount of token$_2$ received x_2 (spent when $x_2 < 0$) given amount of token$_1$ spent x_1 (received when $x_1 < 0$) can be calculated following the steps below:

$$\begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \frac{\sqrt{\frac{4\mathcal{C}\left(\frac{\mathcal{C}}{n}\right)^n}{\mathcal{A} \cdot \prod_{k \neq 2}'} + \left[\left(1 - \frac{1}{\mathcal{A}}\right)\mathcal{C} - \sum_{k \neq 2}'\right]^2} + \left(1 - \frac{1}{\mathcal{A}}\right)\mathcal{C} - \sum_{k \neq 2}'}{2} \\ x_2 &= r_2 - r_2' \end{aligned} \tag{46}$$

where

$$\prod_{k \neq 2}' = r_1' \cdot \prod_{k \neq 1,2} r_k \quad \text{and} \quad \sum_{k \neq 2}' = r_1' + \sum_{k \neq 1,2} r_k \tag{47}$$

d) Slippage: As illustrated in Figure 4c, the slippage that a Curve user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$\begin{aligned} S(x_1) &= \frac{x_1/x_2}{{}_1E_2} - 1 \\ &= \frac{\frac{x_1 \cdot \left[\mathcal{A} \cdot r_1 \cdot \prod_k r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]}{r_1 \cdot \left[\mathcal{A} \cdot r_2 \cdot \prod_k r_k + \mathcal{C} \cdot \left(\frac{\mathcal{C}}{n}\right)^n\right]}}{1 - \frac{\sqrt{\frac{4\mathcal{C}\left(\frac{\mathcal{C}}{n}\right)^n}{a \cdot \prod_{k \neq 2}'} + \left[\left(1 - \frac{1}{a}\right)\mathcal{C} - \sum_{k \neq 2}'\right]^2} + \left(1 - \frac{1}{a}\right)\mathcal{C} - \sum_{k \neq 2}'}{2r_2}} - 1 \end{aligned} \tag{48}$$

e) Divergence loss: Curve's divergence loss in full form cannot be easily presented in a concise and comprehensible fashion. Therefore, for Curve, we use the generalized method to calculate its divergence loss as described in III-C. The divergence loss in the case of a 2-asset pool is presented in Figure 5c.

5) DODO:

[8] Note that $\mathcal{A}$ here is equivalent to $A \cdot n^n$ in Curve's white paper [10].

a) Spot exchange rate: As presented in the previous sections, conventional AMMs derive the exchange rate between two assets in a pool purely from the conservation function. DODO does it the other way around: resorting to external market data as a major determinant of the exchange rate, DODO has its conservation function derived from its exchange rate formula.

The exchange rate between the two assets in a DODO pool is set by the market rate with an adjustment based on the pool composition. We denote the market exchange rate as P, namely 1 token$_2$ $= P$ token$_1$, and the initial reserve for token$_1$ and token$_2$ as $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. The formula Equation 49 sets the exchange rate ${}_1E_2$ higher than the market rate P—i.e. token$_2$ exhibits higher price in the pool than in the market, when the reserve of token$_1$ r_1 exceeds its initial state $\mathcal{C}_1$, and sets ${}_1E_2$ lower than P—i.e. token$_1$ more expensive than its market value, when r_1 falls short of $\mathcal{C}_1$. Formally,

$$ {}_1E_2 = \begin{cases} P\left[1+\mathcal{A}\left(\left(\frac{\mathcal{C}_2}{r_2}\right)^2-1\right)\right], & r_1 \geq \mathcal{C}_1 \\ P \Big/ \left[1+\mathcal{A}\left(\left(\frac{\mathcal{C}_1}{r_1}\right)^2-1\right)\right], & r_1 \leq \mathcal{C}_1 \end{cases} \tag{49} $$

b) Conservation function: DODO's conservation function can be derived from its exchange formula Equation 49. In particular, the initial state of token$_1$ and token$_2$ reserves, $\mathcal{C}_1$ and $\mathcal{C}_2$ can be regarded as the two invariants of the conservation function. This aligns with the definition according to our framework (Section III), as $\mathcal{C}_1$ and $\mathcal{C}_2$ remain constant with swapping activities, but get updated with liquidity provision or withdrawal.

$$ \begin{aligned} r_1 - \mathcal{C}_1 &= \int_{r_2}^{\mathcal{C}_2} P\left[1+\mathcal{A}\left(\left(\tfrac{\mathcal{C}_2}{\delta}\right)^2-1\right)\right] d\delta \\ &= P\cdot(\mathcal{C}_2 - r_2)\cdot\left[1+\mathcal{A}\cdot\left(\tfrac{\mathcal{C}_2}{r_2}-1\right)\right], \quad r_1 \geq \mathcal{C}_1 \end{aligned} \tag{50} $$

$$ \begin{aligned} r_2 - \mathcal{C}_2 &= \int_{r_1}^{\mathcal{C}_1} \frac{1+\mathcal{A}\left(\left(\frac{\mathcal{C}_1}{\delta}\right)^2-1\right)}{P} d\delta \\ &= \frac{(\mathcal{C}_1 - r_1)\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_1}{r_1}-1\right)\right]}{P}, \quad r_1 \leq \mathcal{C}_1 \end{aligned} \tag{51} $$

In the special case of $\mathcal{A} = 1$, when $C_1 = P \cdot C_2$, i.e. liquidity provided on both assets are of equal value, then DODO's conservation function is equivalent to Uniswap, with $r_1 \cdot r_2 = C_1 \cdot P \cdot C_2$. This can be observed from Figure 3, where the DODO's conservation function curve with $\mathcal{A} \to 1$ appears identical to that of Uniswap.

c) Swap amount: The swap amount can be derived directly from the DODO conservation function (Equation 50):

$$ \begin{aligned} r_1' &= r_1 + x_1 \\ r_2' &= \begin{cases} \frac{\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A}) + \sqrt{\left[\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})\right]^2 + 4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2}}{2P\cdot(1-\mathcal{A})}, & r_1' \geq \mathcal{C}_1 \\ \mathcal{C}_2 + \frac{(\mathcal{C}_1 - r_1')\cdot\left[1+\mathcal{A}\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]}{P}, & r_1' \leq \mathcal{C}_1 \end{cases} \\ x_2 &= r_2 - r_2' \end{aligned} \tag{52} $$

d) Slippage: As illustrated in Figure 4d, the slippage that a DODO user experiences when swapping x_1 token$_1$ with x_2 token$_2$ can be expressed as:

$$ S(x_1) = \begin{cases} \frac{2\cdot(1-\mathcal{A})\cdot x_1}{r_1' - \mathcal{C}_1 + \mathcal{C}_2\cdot P -} - 1, & r_1' \geq \mathcal{C}_1 \\ \sqrt{\left[\mathcal{C}_1 - r_1' + P\cdot\mathcal{C}_2\cdot(1-2\mathcal{A})\right]^2 + 4\mathcal{A}\cdot(1-\mathcal{A})\cdot(P\cdot\mathcal{C}_2)^2} & \\ \frac{x_1}{(r_1' - \mathcal{C}_1)\cdot\left[1+A\cdot\left(\frac{\mathcal{C}_1}{r_1'}-1\right)\right]} - 1, & r_1' \leq \mathcal{C}_1 \end{cases} \tag{53} $$

e) Divergence loss: DODO eliminates the kind of divergence loss seen in previously discussed protocols by not forcing liquidity providers to deposit tokens in pre-defined ratios. This is achieved by leveraging price oracles, which allow liquidity providers to deposit an arbitrary combination of base and quote tokens, unlocking single-token exposure. The price oracle is used to protect LPs against heavy arbitrage.

REFERENCES

[1] L. Zhou, K. Qin, C. Ferreira Torres, D. V. Le, A. Gervais, C. F. Torres, D. V. Le, and A. Gervais, "High-Frequency Trading on Decentralized On-Chain Exchanges," 9 2020. [Online]. Available: http://arxiv.org/abs/2009.14021

[2] A. Evans, "Liquidity Provider Returns in Geometric Mean Markets," 6 2020. [Online]. Available: http://arxiv.org/abs/2006.08806

[3] F. Martinelli and N. Mushegian, "Balancer: A non-custodial portfolio manager, liquidity provider, and price sensor," 2019. [Online]. Available: https://balancer.finance/whitepaper/

[4] Bancor, "Bancor V2.1 Technical Explainer," 2020. [Online]. Available: https://drive.google.com/file/d/16EY7FUeS4MXnFjSf-KCgdE-Xyj4re27G/view

[5] CryptoLocally, "GIV Balancer Listing and Staking Rewards Updates," 2020. [Online]. Available: https://cryptolocally.medium.com/giv-balancer-listing-and-staking-rewards-updates-81ebb5843e58

[6] L. De Giglio, "Geyser: Staking Rewards For Uniswap Liquidity Providers," 2021. [Online]. Available: https://medium.com/trips-community/geyser-staking-rewards-for-uniswap-liquidity-providers-115afc6f5c07

[7] DODO Team, "DODO – A Next-Generation On-Chain Liquidity Provider Powered by Pro-active Market Maker Algorithm," 2020. [Online]. Available: https://dodoex.github.io/docs/docs/whitepaper/

[8] H. Andersson, "mStable — Introducing Constant Sum Bonding Curves for Tokenised Assets," 2020. [Online]. Available: https://medium.com/mstable/introducing-constant-sum-bonding-curves-for-tokenised-assets-6e18879cdc5b

[9] Uniswap, "Flash Swaps," 2020. [Online]. Available: https://uniswap.org/docs/v2/core-concepts/flash-swaps/

[10] M. Egorov, "StableSwap-efficient mechanism for Stablecoin liquidity," 2019.

[11] D. Senchenko, "Impermanent Losses in Uniswap-Like Markets," 2020. [Online]. Available: https://dsenchenko.medium.com/impermanent-losses-in-uniswap-like-markets-4315359ea9b1

[12] Ethereum, "Types," 2020. [Online]. Available: https://docs.soliditylang.org/en/latest/types.html

[13] G. Angeris, A. Evans, and T. Chitra, "Replicating Market Makers," 3 2021. [Online]. Available: http://arxiv.org/abs/2103.14769

[14] Sushiswap, "The SushiSwap Project," 9 2020. [Online]. Available: https://sushiswapchef.medium.com/the-sushiswap-project-dd6eb80c6ba2

[15] QuickSwap, "QuickSwap," 2021. [Online]. Available: https://quickswap.exchange/#/swap

[16] Polygon, "Polygon — Ethereum's Internet of Blockchains," 2021. [Online]. Available: https://polygon.technology/

[17] QuickSwap, "QuickSwap Info," 4 2021. [Online]. Available: https://info.quickswap.exchange/home

[18] E. Hertzog, G. G. G. Benartzi, and G. G. G. Benartzi, "Bancor Protocol Continuous Liquidity for Cryptographic Tokens through their Smart Contracts," 2018. [Online]. Available: https://storage.googleapis.com/website-bancor/2018/04/01ba8253-bancor_protocol_whitepaper_en.pdf

[19] Bancor, "Announcing Bancor V2," 4 2020. [Online]. Available: https://blog.bancor.network/announcing-bancor-v2-2f56b515e9d8

[20] Bancor Network, "FAQs - Bancor Network," 2021. [Online]. Available: https://docs.bancor.network/faqs#how-does-impermanent-loss-insurance-work

[21] H. Adams, "Uniswap Whitepaper (v1)," 2018. [Online]. Available: https://hackmd.io/C-DvwDSfSxuh-Gd4WKE_ig

[22] P. Baker, "DeFi Project bZx Exploited for Second Time in a Week, Loses $630K in Ether," 2 2020. [Online]. Available: https://www.coindesk.com/defi-project-bzx-exploited-for-second-time-in-a-week-loses-630k-in-ether

[23] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 Core," 2020.

[24] Harvest Finance, "Harvest Flashloan Economic Attack Post-Mortem ," 10 2020. [Online]. Available: https://medium.com/harvest-finance/harvest-flashloan-economic-attack-post-mortem-3cf900d65217

[25] B. Pirus, "Cheese Bank's multi-million-dollar hack explained by security firm," 11 2020. [Online]. Available: https://cointelegraph.com/news/cheese-bank-s-multi-million-dollar-hack-explained-by-security-firm

[26] PeckShield, "Value DeFi Incident: Root Cause Analysis," 11 2020. [Online]. Available: https://peckshield.medium.com/value-defi-incident-root-cause-analysis-fbab71faf373

[27] M. Young, "Warp Finance reportedly loses up to $8M in flash loan attack," 12 2020. [Online]. Available: https://cointelegraph.com/news/warp-finance-reportedly-loses-up-to-8m-in-flash-loan-attack

[28] H. Adams, N. Zinsmeister, M. Salem moody, u. River Keefer, and D. Robinson, "Uniswap v3 Core," 2021.

[29] 1inch Network, "Balancer Pool with STA Deflationary Token Incident," 2020. [Online]. Available: https://blog.1inch.io/balancer-hack-2020-a8f7131c980e

[30] F. Martinelli, "Introducing Balancer V2: Generalized AMMs," 2 2021. [Online]. Available: https://medium.com/balancer-protocol/balancer-v2-generalizing-amms-16343c4563ff

[31] B. Dale, "SushiSwap Will Withdraw Up to $830M From Uniswap Today: Why It Matters for DeFi," 9 2020. [Online]. Available: https://www.coindesk.com/sushiswap-uniswap-migration-defi-amm-wars

[32] A. Bukov and M. Melnik, "Mooniswap by 1inch.exchange," 2020.

[33] Kyber Network, "Kyber 3.0: Architecture Revamp, Dynamic MM, and KNC Migration Proposal," 2021. [Online]. Available: https://blog.kyber.network/kyber-3-0-architecture-revamp-dynamic-mm-and-knc-migration-proposal-acae41046513

[34] michaelhly, macalinao, and joncinque, "michaelhly/stable-swap-program: StableSwap program for the Solana blockchain," 2021. [Online]. Available: https://github.com/michaelhly/stable-swap-program

[35] TrueSwap, "Introducing Trueswap," 3 2021. [Online]. Available: https://trueswapfinance.medium.com/introducing-trueswap-f3438e32b568

[36] HydraDX, "Intro — HydraDX Docs," 2021. [Online]. Available: https://docs.hydradx.io/

[37] Gyroscope Finance, "Gyroscope, the new all-weather stablecoin — Gyroscope Protocol," 2021. [Online]. Available: https://gyro.finance/

[38] ——, "Gyroscope AMMs - Gyroscope Protocol," 2021. [Online]. Available: https://docs.gyro.finance/learn/gyro-amms

[39] EulerBeats, "EulerBeats — About," 2021. [Online]. Available: https://eulerbeats.com/about

[40] Pods Finance, "Pods Finance — The easiest way to hedge crypto," 2021. [Online]. Available: https://www.pods.finance/

[41] Balancer, "Liquidity Bootstrapping Pool - Balancer," 2021. [Online]. Available: https://docs.balancer.finance/guides/smart-pool-templates-gui/liquidity-bootstrapping-pool

[42] A. Niemerg, D. Robinson, and L. Livnev, "YieldSpace: An Automated Liquidity Provider for Fixed Yield Tokens," 2020. [Online]. Available: https://yield.is/Yield.pdf

[43] D. Robinson and A. Niemerg, "The Yield Protocol: On-Chain Lending With Interest Rate Discovery," Tech. Rep., 2020.

[44] Notional Finance, "Notional Finance," 2021. [Online]. Available: https://notional.finance/

[45] ——, "Notional AMM," 2020. [Online]. Available: https://docs.notional.finance/traders/technical-topics/notional-amm

[46] Gnosis, "Custom Market Maker · Gnosis Developer Portal Gnosis Protocol," 2020. [Online]. Available: https://docs.gnosis.io/protocol/docs/intro-cmm/

[47] jakub, "What is a Vampire Attack? SushiSwap Saga Explained," 2020. [Online]. Available: https://finematics.com/vampire-attack-sushiswap-explained/

[48] Blank, "Blank features beyond basic privacy (#2): Protecting your IP in DeFi," 2021. [Online]. Available: https://blankwallet.medium.com/blank-features-beyond-basic-privacy-2-protecting-your-ip-in-defi-11bc76f2d67b

[49] C. Kisagun, "Preventing DEX Front-running with Enigma," 2019. [Online]. Available: https://blog.enigma.co/preventing-dex-front-running-with-enigma-df3f0b5b9e78

[50] K. Qin, L. Zhou, and A. Gervais, "Quantifying Blockchain Extractable Value: How dark is the forest?" 1 2021. [Online]. Available: http://arxiv.org/abs/2101.05511

[51] K. Qin, L. Zhou, B. Livshits, and A. Gervais, "Attacking the DeFi Ecosystem with Flash Loans for Fun and Profit," Tech. Rep., 2020. [Online]. Available: http://arxiv.org/abs/2003.03810

[52] Y. Cao, C. Zou, and X. Cheng, "Flashot: A Snapshot of Flash Loan Attack on DeFi Ecosystem," 1 2021. [Online]. Available: http://arxiv.org/abs/2102.00626

[53] D. Perez, S. M. Werner, J. Xu, and B. Livshits, "Liquidations: DeFi on a Knife-edge," 2020.

[54] D. Wang, S. Wu, Z. Lin, L. Wu, X. Yuan, Y. Zhou, H. Wang, and K. Ren, "Towards understanding flash loan and its applications in defi ecosystem," 10 2020. [Online]. Available: http://arxiv.org/abs/2010.12252

[55] F. Victor and A. M. Weintraud, "Detecting and Quantifying Wash Trading on Decentralized Cryptocurrency Exchanges," p. 10, 2 2021. [Online]. Available: http://dx.doi.org/10.1145/3442381.3449824

[56] L. Gudgeon, D. Perez, D. Harz, B. Livshits, and A. Gervais, "The Decentralized Financial Crisis," in *Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 6 2020, pp. 1–15. [Online]. Available: https://ieeexplore.ieee.org/document/9150192/

[57] G. Angeris, A. Evans, and T. Chitra, "A Note on Privacy in Constant Function Market Makers," 3 2021. [Online]. Available: http://arxiv.org/abs/2103.01193

[58] D. Stone, "Trustless, privacy-preserving blockchain bridges," 2021. [Online]. Available: http://arxiv.org/abs/2102.04660

[59] G. Angeris, H.-T. Kao, R. Chiang, C. Noyes, and T. Chitra, "An analysis of Uniswap markets," 2019.

[60] Y. Lo and F. Medda, "Uniswap and the rise of the decentralized exchange," 11 2020.

[61] G. Angeris and T. Chitra, "Improved Price Oracles: Constant Function Market Makers," in *Advances in Financial Technologies*. New York, NY, USA: ACM, 10 2020, pp. 80–91. [Online]. Available: https://dl.acm.org/doi/10.1145/3419614.3423251

[62] M. B. Garman, "Market microstructure," *Journal of Financial Economics*, vol. 3, no. 3, pp. 257–275, 6 1976. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0304405X76900064

[63] W. Perraudin and P. Vitale, "Interdealer Trade and Information Flows in a Decentralized Foreign Exchange Market," in *The Microstructure of Foreign Exchange Markets*. University of Chicago Press, 1996, pp. 73–106.

[64] F. Nava, "Efficiency in decentralized oligopolistic markets," *Journal of Economic Theory*, vol. 157, pp. 315–348, 5 2015. [Online]. Available: www.sciencedirect.comwww.elsevier.com/locate/jet

[65] S. Malamud and M. Rostek, "Decentralized Exchange," *American Economic Review*, vol. 107, no. 11, pp. 3320–3362, 11 2017. [Online]. Available: https://pubs.aeaweb.org/doi/10.1257/aer.20140759http://pubs.aeaweb.org/doi/10.1257/aer.20140759

[66] R. Hanson, "Combinatorial Information Market Design," *Information Systems Frontiers*, vol. 5, no. 1, pp. 107–119, 2003. [Online]. Available: http://hanson.gmu.edu

[67] ——, "Logarithmic Markets Scoring Rules for Modular Combinatorial Information Aggregation," *The Journal of Prediction Markets*, vol. 1, no. 1, pp. 3–15, 12 2012. [Online]. Available: http://www.bjll.org/index.php/jpm/article/view/417

[68] A. Othman, D. M. Pennock, D. M. Reeves, and T. Sandholm, "A Practical Liquidity-Sensitive Automated Market Maker," *ACM Transactions on Economics and Computation*, vol. 1, no. 3, pp. 1–25, 9 2013. [Online]. Available: https://dl.acm.org/doi/10.1145/2509413.2509414

[69] A. Brahma, M. Chakraborty, S. Das, A. Lavoie, and M. Magdon-Ismail, "A bayesian market maker," in *Proceedings of the ACM Conference on Electronic Commerce*. New York, New York, USA: ACM Press, 2012, p. 215. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2229012.2229031

[70] J. Jumadinova and P. Dasgupta, "A Comparison of Different Automated Market-Maker Strategies," 2012, pp. 2009–2012. [Online]. Available: http://www.cs.allegheny.edu/~jjumadinova/market-maker_AMEC.pdf
[71] C. Slamka, B. Skiera, and M. Spann, "Prediction Market Performance and Market Liquidity: A Comparison of Automated Market Makers," *IEEE Transactions on Engineering Management*, vol. 60, no. 1, pp. 169–185, 2 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6189381/
[72] Y. Wang, "Automated market makers for decentralized finance (DeFi)," 9 2020. [Online]. Available: http://arxiv.org/abs/2009.01676
[73] J. Peterson and J. Krug, "Augur: a decentralized, open-source platform for prediction markets," 2015.
[74] A. Capponi and R. JIA, "The Adoption of Blockchain-based Decentralized Exchanges: A Market Microstructure Analysis of the Automated Market Maker," 2021. [Online]. Available: https://ssrn.com/abstract=3805095