The Case for Stochastically Dynamic AMMs

Kristof Lommers[1] Jack Kim[2] Boris Skidan[2]
Viktor Smits[2] *†

April 10, 2023

Abstract

AMMs represent a class of decentralized exchange mechanisms that rely on a fixed mathematical formula to price assets. Popular AMM protocols utilize a constant function approach which is deterministic in design. However, these designs are plagued with issues such as impermanent loss, slippage costs, and market risk incomputability. More specifically, the design allows professional arbitrage traders to siphon off value from liquidity providers. We would like to make the case for the development of a new class of AMM designs based on stochastic pricing that would dynamically adjust to market information. The stochastic design would function like a profit maximizing market maker which we could solve through optimal control theory.

1 Introduction

Exchanges within the decentralized finance (DeFi) ecosystem have mainly utilized automated market maker (AMM) designs to facilitate trades. As of the moment of writing AMMs handle approximately 15% of trading volume in crypto currencies with an approximate trading volume of $1 trillion in. AMMs represent a class of decentralized exchange mechanisms that rely on a fixed mathematical formula to price assets. Liquidity pools are created for a pair or basket of assets where trades are made against this pool according to a pre-specified function. Currently popular AMM protocols utilize a constant function approach which is deterministic in design. As a result of this design, we argue that they are plagued with issues such as deterministic slippage cost, market risk incomputability, and impermanent loss. These issues result in a relatively high price impact of trades for users and relatively high fees to compensate liquidity providers for the associated risks. More

*1 HashCurve, UCL Centre for Blockchain Technologies
†2 HashCurve

specifically, the design allows arbitrageurs to siphon off value from traders and liquidity providers in these AMM protocols.

In this article we would like to make the case for an AMM design using stochastic bonding curves. In this case the pricing would be dynamic based on market circumstances according to a profit maximizing market maker. Under certain assumptions, the solution can be approximated in tractable form. In this construct one is essentially solving for an optimal control problem as one is deciding on the market maker's optimal actions in terms of his profit and loss statement. This design would help to mitigate various problems in current AMM mechanisms such as decreasing price impact of trades, more efficient use of provided capital, and decreased impermanent loss for liquidity providers.

We discuss various implementation considerations and possible design choices. One major protocol design consideration is the specific use case of the AMM. It should be emphasized that in this paper we present the general case for stochastic AMMs where there are numerous design considerations — both in the stochastic model and the protocol — based on the use case. AMMs have been used to swap tokens, market make perpetual futures, set rates derivatives, etc. We present a general case where the model would need to be tuned for the specific use case. This is the first paper in the HashCurve series on Stochastic AMMs where HashCurve [20] has solved for a stochastic AMM for perpetual futures. This paper contributes to the growing body of literature on AMMs and more generally on the use of bonding curves in DeFi. We propose a novel approach to bonding curve by making them stochastic which introduces more dynamic pricing on market developments.

2 Background on AMMs

Exchanges within the DeFi ecosystem have mainly utilized AMM designs to facilitate trades. AMM protocols allow traders to swap assets against the liquidity pool which is a smart contract holding the assets in question with specific rules to deposit or withdraw assets. Liquidity providers deposit the relevant assets in the liquidity pools and receive a fee for this service. When people swap assets with the protocol they are trading against a pool of liquidity provided by LPs and managed by the AMM smart contract.

This is largely in contrast to the traditional mechanism in which investors trade against a specific counterparty on a (centralized) exchange. In this setup investors can trade assets without the need of a centralized intermediary as it constitutes a smart contract on the distributed ledger or a system of custody. The task of the AMM is to adjust prices to demand and supply information from market participants. Instead of matching buy and sell orders, AMMs determine asset price algorithmically through a bonding function. This bonding function allows the exchange rates to move along predefined trajectories which are conditioned upon the quantities of available assets. It is important to note that AMMs do not update their prices based on new information, and that price efficiency is de facto obtained through arbitrage activity.

AMMs were initially developed for information aggregation for markets where payoffs depend on a specific future state such as prediction markets [4]. Various formulations of automated market makers have been proposed, such as market scoring rules, constant-utility market maker and constant function market maker [1]. Constant function market makers (CFMMs) have been the first class of AMMs to be specifically applied to financial markets while other designs have been rather used in applications such as prediction markets. They were used to construct DEXs for crypto and constitute the most popular AMM construct in DeFi [4]. The bonding curve of these AMMs constitutes a constant function where the combined asset reserves of trading pairs must be kept constant.

Existing bonding curve designs in this category include constant product market maker, constant sum market maker, constant mean market maker or hybrid versions consisting of a combination of the aforementioned ones. The most popular CFMM has been the constant product design where the product of the reserves of the assets within the liquidity pool has to be constant. It is important to note that CFMMs are deterministic in the design of their bonding curve. Popular AMM protocols by trading volume – such as Uniswap, Sushiswap, Balancer, Curve, etc. – are based on a constant function bonding curve. We would like to refer to Xu et al. [34] for an extensive overview and discussion of existing AMM designs.

Finally, AMMs have been used for numerous other use cases within the DeFi ecosystem. The AMM construct has been mainly popularized in token swap protocols, but has been adapted to several DeFi applications such as crypto options (e.g., Panoptic), crypto futures (e.g., Perpetual Protocol), rate swaps (e.g., Voltz), and NFT exchanges (e.g., Caviar). It should be noted that in the case of derivatives, a virtual AMM (vAMM) model has been applied which is slightly different to the conventional AMM as users trade against a pool where they have to store collateral in the smart contract based vault. There are no liquidity providers in this case as the liquidity comes from the vault that is funded by traders. It is virtual in the sense that there is no proper asset pool backing the counterparty risk and it is undercollateralized. The concepts discussed in this paper could be applied — although slightly differently — to the vAMM construct as well as it functions as an automated market where the pricing depends on bonding curves.

3 AMM Design Weaknesses

Angeris and Chitra [2] find that under sufficient conditions and under fairly general assumptions, agents who interact with constant function market makers are incentivized to correctly report the price of an asset in a computationally efficient way. They provide sufficient conditions for these CFMMs to be well-behaved, in the sense that agents are incentivized to correctly report asset prices and can never drain the assets of the CFMM by only trading with the given CFMM. The CFMM constitutes thus a sufficient functioning mechanism for (decentralized) trading purposes. There are, however, a few core problems including deterministic slippage costs, impermanent loss for liquidity providers, and improper accounting of market risk. These problems result in higher price impact of trades and increased fees (to compensate liquidity providers for the risk). More specifically, the problems contribute to the siphoning of value from AMM traders and liquidity providers to arbitrageurs.

The primary risk of liquidity providers is impermanent loss (IL) which constitutes the difference in value over time between depositing tokens in an AMM versus simply holding those assets. This loss occurs when the market-wide price of tokens inside an AMM diverges in any direction. Since AMMs do not automatically adjust their exchange rates, an arbitrageur is required to buy the underpriced assets or sell the overpriced assets until the prices offered by the AMM match the market-wide price of external markets. The profit extracted by arbitrageurs is siphoned from the pockets of liquidity providers, creating a loss. LP positions in current AMM designs are de facto short volatility investments. Current AMM designs compensate liquidity providers for this risk of impermanent loss by relatively high fees. However, Loesh et al. [25] find that most passive LPs in Uniswap – except for professionally active "flash LPs" who provide intra-block liquidity – lost money relative to just holding the asset due to impermanent loss. Implementing range orders only partially solves this problem as the impermanent loss is just capped at the lower end of the range. Numerous others such as 0xfbifemboy [**?**], thiccythot and charliemktplace have performed analysis on Uniswap LPs and found that they were losing money.

The price, which is determined by the bonding curve, follows a deterministic function and as a result slippage costs follow a deterministic function as well. Prices are a (constant) function of the amount of tokens and are thus relatively deterministically predictable based on the inventories. From the lack of dynamic price adjustments AMMs could be seen as rather "static". This deterministic slippage leads to a deadweight loss that arbitrageurs can capture at the expense of users [13]. This deterministic slippage cost could also lead to front-running and miner extractable value [12]. Furthermore, conventional AMM designs do not properly account for market risk as price discovery happens in a rather naive way. AMM designs de facto assume price arbitrage to make AMM prices efficient. More specifically, the price discovery depends on cross-exchange arbitrage rather than a pricing oracle for reference prices. The market risk is paid by the users and liquidity providers and there is no proper computation of market risk within the pricing of the AMM mechanism. It should be noted that this market risk (incomputability) creates a hedging difficulty.

Finally, the vAMM construct inherits similar weaknesses as a result from its deterministic bonding curve design. Some have argued that due to path independence, vaults have enough collateral to renumerate traders. However, this assumes that undercollateralized positions would be liquidated on time before losing money which is not always the case during market volatility. Because of this undercollateralization most vAMMs have an insurance pool to pay for these losses. You could argue that the vAMM is a victim of adverse selection by traders in this case. The vAMM inherits the same impermanent loss as the conventional AMM construct only the traders would bear the costs instead of LPs.

4 Why Stochastically Dynamic AMMs?

AMMs have been a popular trading venue as they have the advantage of decentralization and continuous liquidity. Traditionally market makers provide inventory and help determine prices by matching supply and demand in an auction-like game. In AMMs, liquidity providers supply the asset inventory and the prices are determined by the pricing algorithm. The advantage of this is that users obtain immediate liquidity without having to find an exchange counterparty, whereas liquidity providers receive fees for this service. AMMs allow for an exchange to occur immediately which could be important for low liquidity assets. This makes the AMM design interesting for relatively new protocols to launch crypto tokens while allowing for a simple trading environment without the need of active market makers to provide liquidity. Traders have an disincentive to post orders in thin markets as they would reveal information with little benefit which relates to the no-trade theorem of Milgram and Stokey [11].

You could view market making as a series of actions where an equilibrium is found that balances supply and demand [15]. The rules of the auction are fixed by the exchange, and in the case of AMMs these rules are executed by a smart contract. Although there are some issues in current AMM designs, tools within Web3 provide an opportunity for built-in mechanism design within smart contracts. One can argue that the biggest innovation of smart contracts was the ability to enforce design mechanisms and thus directly implement market design. The goal of market design is to help reduce some of the negative externalities and inefficiencies preventing markets from achieving efficient first best outcomes [22]. As argued by Robinson and Konstantopoulos [28] Ethereum is a dark forest as it is an adversarial environment where code is law and weak designs are exploited. We argue that you could directly implement more advanced mechanism design considerations through smart contracts, and thus solve the current problems in AMMs through old-fashioned mechanism design.

To understand how a stochastic design could help with some of the problems in AMMs, we would need to take a closer look at impermanent loss. Milionis et al. [26] define impermanent loss in different ways: as cost of commitment for giving up future optionality, as the cost of arbitrage against the pool, and as an information cost due to the unavailability of correct market prices. For simplicity, we could view impermanent loss as the loss of liquidity providers because external arbitrageurs harness gamma convexity gains on every round trip in a self-financed arbitrage strategy against the AMM. It should be noted that the gains of external arbitrageurs are stochastic in this case because different paths ending up at the same spot go through different realized volatility. In a limit order book market makers would dynamically adjusts their hedge portfolio for a given percentage move in the spot and harnesses the cash gamma. AMMs have generally failed in protecting users from toxic order flow of arbitrageurs where especially unsophisticated users are taken advantage of. The evidence tends to show that passive LPs have received the short end of the stick. As argued by Cohen (2022), providing ambient liquidity (liquidity that is not very actively managed) on volatile tokens is a negative expected value strategy in AMMs.

We would argue that the AMM mechanism would need to manage the positions while allowing LPs to provide passive ambient liquidity. You could apply a stochastic framework in an optimal control problem that decides on the market maker's optimal actions in terms of quoting prices. This way the effective bonding curve would be more dynamic on market circumstances. In this construct one is essentially solving for an optimal control problem as one is deciding on the market maker's optimal actions in terms of his expected PnL, under certain risk constraints. This design would allow for passive LP participation where the liquidity is managed by the AMM protocol. A stochastic design tackles the previously discussed problems in AMMs. The impermanent loss for liquidity providers is solved through the stochastic mechanism that actively manages LP positions. The slippage costs adjust by order size and actions like front-running of trades can be mitigated. Market risk is taken into account in the market oracle based pricing mechanism as prevailing market conditions such as volatility impact the MM behavior. Stochastic bonding curves can improve price discovery by allowing the market to adapt to changes in supply and demand in a more responsive way.

5 Stochastic Models for Optimal Market Maker Behavior

One could construct a stochastic framework to derive a solution that describes optimal behavior of a market maker. As aforementioned, one would be solving an optimal control problem in deciding the market maker's optimal actions or policies. This is similar to the work of Chitra et al. [12] who discuss DeFi mechanisms from an optimal control point of view through a stochastic model. The market maker aims to maximize his expected period profit-and-loss (PnL) subject to various (risk) constraints. There are other approaches to this problem such as one implemented by Swaap Labs [5] where they have designed a constant geometric AMM which embeds a stochastic spread mechanism. We would argue, however, that one needs to incorporate a framework that solves for optimal market making behavior. There is substantial previous work on stochastic models for optimal market making behavior including the seminal paper of [3]. Other authors have extended the model to incorporate inventory constraints ([17]), directional alpha views ([14]), mutually exciting order arrivals ([32], [17]), and regime switching ([24]). These frameworks can be applied to derive optimal market making behavior in AMM protocol designs.

We would first need to define the price dynamics of the spot process where you would need to assume a distribution which allows to construct a stochastic process with tunable moments. For example, Kim et al. [19] in HashCurve protcol assume that the joint price and volatility dynamics are given by a Schobel-Zhu model. In addition to the price dynamics we need to specify the dynamics of order arrivals. For example, Kim et al. [19] in HashCurve protocol use the setup of [3] and [**?**] with two independent Poisson processes $N^{\pm}$ for bids and asks, with intensities $\lambda^{\pm}$ respectively. This model operates under the simplifying assumptions of no direct (self and mutually exciting) feedback of orders on the price as addressed in [9] or [**?**]. However, the impact of orders, acceleration, momentum, and imbalances can be expressed indirectly through α_t and the stochastic volatility enables volatility clustering. For order arrival dynamics Kim et al. [19] follow the classical set up as in [**?**]. The market maker controls her ask and bid quotes, $p^+(t)$ and $p^-(t)$ which can be expressed as non-negative spreads, $\delta^+(t) = p^+(t) - S(t)$ and $\delta^-(t) = S(t) - p^-(t)$. They assume fill rates of the form $e^{-\gamma_t^{\pm}\delta_t^{\pm}}$. Implicitly the alpha process depends on all market variables including order arrivals $\alpha = A(t, S_t, N_t^+, N_t^-)$. The inventory of the market maker is given by

$$Q_t = N_t^- - N_t^+ \tag{1}$$

while the market maker's cash evolves by

$$\begin{aligned} dX_t &= p_t^+ dN_t^+ - p_t^- dN_t^- \\ &= (S_t + \delta_t^+) dN_t^+ - (S_t - \delta_t^-) dN_t^-. \end{aligned} \tag{2}$$

Finally you would need to define the market maker's utility function in terms of his PnL. The market maker's goal would be to maximize his utility function given cash X_t, inventory q_t, and price process S_t, $U(X_T, S_T; q_T)$ (and potentially other relevant variables). For example, Kim et al. [19] in HashCurve protcol assume linear utility over horizon T with quadratic inventory penalization which would give

$$U(X_T, S_T, Q_T) = X_T + Q_T S_T - \eta Q_T^2 \tag{3}$$

where $\eta \geq 0$ represents the risk-aversion parameter regarding residual inventory. In this formulation the market maker wishes to maximize his MtM wealth, but is penalized for carrying over inventory. The associated value function would be

$$v(t, x, s, q) = \mathbf{E}_{t,x,s,q} \left[X(T) + Q(T)S(T) - \eta Q^2(T) \right] \tag{4}$$

The utility function can take on other forms (monotonic in wealth and concave). This would represent the market maker's problem which you would like to optimize where you could use stochastic optimization methods. Stochastic optimization deals with the challenge of finding optimal decisions in dynamic systems affected by random noise. Such systems are often modeled by stochastic differential equations (SDEs) or controlled stochastic processes. A central tenet of optimal control theory is the use of optimization techniques to determine the control input that minimizes or maximizes a certain performance criterion, typically expressed as a cost or utility function. An optimal control problems typically involve a dynamic system, which can be described by a state equation and a control input. The objective of the problem is to find the control input that optimizes a certain performance criterion, such as minimizing the cost of operation or maximizing the system's efficiency. Mathematically, an optimal control problem can be formulated as follows: Given a dynamic system described by the state equation:

$$\frac{dx(t)}{dt} = f(x(t), u(t), t), \tag{5}$$

where $x(t)$ is the state vector, $u(t)$ is the control input, and f is a function describing the system's dynamics. The goal is to find a control $u^*(t) = (\delta^-, \delta^+)$ that maximizes (or minimizes depending on orientation of objective function) a performance criterion J, given by an integral:

$$J[u] = \int_{t_0}^{T} g(x(t), u(t), t, x, q)dt + h(x(T), x, q), \tag{6}$$

where g is the running cost, h is the terminal cost, and T is the horizon.

The value function represents the expected cumulative cost or reward from following an optimal control policy, given the current state of the system. In optimal control theory, these can be solved by deriving the Hamilton-Jacobi-Bellman (HJB) equations, which are partial differential equations (PDE) that characterize the value function associated with the optimal control problem. In cases where a closed-form solution exists, these are typically solved by guessing an appropriate ansatz, substituting the optimal controls for the verification equation, and solving the equations via Feynmann-Kac representations. However, in most settings and specifications of dynamics, the optimal policy needs to be derived numerically. Yet, under certain assumptions, minimum tick size constraints, closed-form approximations may be amenable.

In the limit the stochastic AMM should thus be efficient if the design solves for the HJB equation which gives a necessary and sufficient condition for optimality. It should be noted that the simplicity comes from approximating the solution by a closed-form solution. For example, Kim et al. [19] have derived for HashCurve protocol analytical approximations for the solution to an optimal market making model, under simplifying assumptions that may be relaxed later as the market evolves. We would like to emphasize, again, that this section provides a generalized discussion and you would need to define the market making problem based on the specific use case.

6 Touching the Surface on AMM Design Considerations

First of all, the main design consideration is based on the specific use case of the AMM. In this paper we presented a general case for a stochastic AMM but the underlying quantitative model and protocol design would differ substantially based on the application in question. Kim et al. [20] introduce a stochastic AMM design for a perpetual futures protocol (for illiquid and novel markets) which requires a completely different protocol design compared to an AMM for token swaps. For example in the case of token swaps you could propose a stochastic AMM design where liquidity is withdrawn when the price escapes the lower end of a computed range. The mechanism would work with that range orders can be set with automatic withdrawal from the liquidity pool when a range bound is hit so that liquidity providers do not have to actively manage their position. By combining a stochastic mechanism that dictates automated liquidity adjustment in computed price ranges you could allow the AMM to de facto function as an active market maker in a synthetic order book. More specifically, you could apply range orders where liquidity is automatically withdrawn from unprofitable price ranges when prices hit the lower end of the stochastic spread. This allows for similar dynamics as seen in order books where liquidity comes in price ranges and is actively moving based on incoming information.

Based on the specified use case different trade-offs should be considered in the implementation. Due to the limitations of blockchains most DeFi protocols have been unable to leverage computational models. You could argue that the main advantage of a closed-form solution is the possibility to compute things on-chain. For example, Swaap Labs [5] — who designed a constant geometric AMM which embeds a stochastic spread mechanism — compute the mechanism on-chain where they use oracles for some of the base variables. However, more advanced models would be — by current blockchain technology standards — too computationally heavy to handle. Since the AMM would require a data oracle one could argue to perform the necessary data pipeline and computations off-chain where the optimal price is fed into the AMM through an oracle. For example, Kim et al. [20] their stochastic AMM design tries to strike a balance with off-chain computations where they provide verification procedures for said computations. You could also decentralize this computation by giving the formula to others to compute the optimal prices. Numerous proposals have been floating around such as one by Suresh, Latif and Shah [**?**] who suggest a system where oracle providers have to stake reputation and collateral. Finally, when using oracles numerous design considerations would need to be taken into account as there are numerous issues such as oracle lag time, reliability, potential oracle manipulation, etc. For example, samczsun [30] discusses oracle vulnerabilities and the need to think them through well in DeFi protocol designs. For example, Kim et al. [21] discuss numerous anti-manipulation measures for NFT index oracles for HashCurve protocol.

On public blockchains, such as Ethereum, traders can view unconfirmed transactions in the mempool and front-run a trade by putting in an order. For example, as discussed in a report from Delphi Digital [10], there are just in time (JIT) liquidity providers extracting fees from LPs. Front-running introduces a wealth transfer from protocol LPs and traders to arbitrageurs — it is essentially a negative externality on the protocol design and introduces behavior that the stochastic model would not take into account. Various solutions have been proposed in this regard such as commit-reveal schemes, systems where unprocessed transactions are kept hidden or batch orders. The need and design for these mechanisms depends on the specific use case, design and the underlying blockchain on which the protocol is build. You could add a stochastic element to the closed-form computation so it becomes more difficult to predict the fill-price. For example, Kim et al. [19] in HashCurve protcol assume a stochastic fill rate which is implement through a double stochastic mechanism where fills are determined using probabilities from the same distribution.

You could also argue for a stochastic AMM that has the base model on-chain where the parameters are tuned according to a stochastic model. AMMs typically employ off-chain governance mechanisms, whereby token holders exercise voting rights to adjust various protocol parameters. You could design a more dynamic and automated system by using a stochastic model for this. The current process of parameter selection remains predominantly manual and is often carried out by the core team behind the protocol. Xu et al. [35] propose a semi-automatic parameter adjustment methodology using reinforcement learning which autonomously generates intuitive, data-driven governance proposals to adjust protocol parameters.

You could propose a similar mechanism where the parameters would be automatically tuned using an underlying stochastic model. It should be noted that using a stochastic model would provide a structured framework through intuitive and empirically backed assumptions. Machine learning, in its ability to learn the underlying structure and properties of the data, tends to be heavily used for financial models. However, it is important to emphasise that there are numerous difficulties in this data exercise in the case of limited data and the volatile non-stationary nature of digital asset prices. The advantage of using closed-form models is that it can improve model interpretability, parameter estimation, a way to think structurally around a problem. A further advantage of the structured models is that they are easier "control" and one can explain the dynamics — at least under the considered assumptions of the model.

We want to emphasize this discussion only touches the surface as there are a lot of use case specific model and protocol design applications, and want to reiterate that this paper is rather a primer on the topic of stochastic AMMs. For example, Jack et al. [19] provide a discussion on a stochastic AMM model and protocol design for perpetual futures where they are developing HashCurve protocol.

7 Conclusion

In conclusion, this paper has presented a novel approach to automated market makers (AMMs) in the burgeoning decentralized finance (DeFi) ecosystem by introducing stochastic bonding curves. By moving away from deterministic AMM designs that are associated with issues such as deterministic slippage cost, market risk incomputability and impermanent loss, our proposed stochastic AMM aims to address these challenges and improve the overall efficiency of the DeFi market. The stochastic AMM design would emulate a profit-maximizing market maker and thus facilitate more passive participation from liquidity providers.

We have discussed numerous implementation considerations and potential design choices for the proposed stochastic AMM, emphasizing the need for customization based on specific use cases. As AMMs have been employed in various capacities, such as token swaps, perpetual futures market making, and rate derivatives, the stochastic model and protocol design must be fine-tuned accordingly. With this paper we hope to incentivize further research and development of stochastic AMMs. As the DeFi ecosystem continues to evolve, we expect that our proposed stochastic AMM framework will necessitate further refinements to accommodate new advancements and address emerging risks. Ultimately, this paper lays the groundwork for exploring the potential of stochastic AMMs and their implications for the future of decentralized finance, paving the way for more efficient and resilient market mechanisms in this rapidly changing domain.

References

[1] Abernethy, J.D., Frongillo, R.M. and Kutty, S., 2015. On risk measures, market making, and exponential families. ACM SIGecom Exchanges, 13(2), pp.21-25.

[2] Angeris, G. and Chitra, T., 2020, October. Improved price oracles: Constant function market makers. In Proceedings of the 2nd ACM Conference on Advances in Financial Technologies (pp. 80-91).

[3] Avellaneda, M., and Stoikov, S. (2008). High-frequency trading in a limit order book. Quantitative Finance, 8(3), 217-224.

[4] Berenzon, D. 2020. Constant Function Market Makers: DeFi's "Zero to One" Innovation. Medium article.

[5] Bouba, D. 2021. Swaap.finance: Introducing the Matrix-MM.

[6] Browne, S., 1995. Optimal investment policies for a firm with a random risk process: exponential utility and minimizing the probability of ruin. Mathematics of operations research, 20(4), pp.937-958.

[7] Budish, E., Cramton, P. and Shim, J., 2014. Implementation details for frequent batch auctions: Slowing down markets to the blink of an eye. American Economic Review, 104(5), pp.418-24.

[8] Capponi, A. and Jia, R. 2021. The Adoption of Blockchain-based Decentralized Exchanges: A Market Microstructure Analysis of the Automated Market Maker. Available at SSRN 3805095.

[9] Cartea, A., and Jaimungal, S. (2015). Optimal execution with limit and market orders. Quantitative Finance, 15(8), 1279-1291.

[10] Charbonneau, J. 2022. MEV Manifesto. Delphi Labs.

[11] Chen, Y. and Pennock, D.M., 2012. A utility framework for bounded-loss market makers. arXiv preprint arXiv:1206.5252.

[12] Chitra, T., Kulkarni, K., Angeris, G., Evans, A. and Xu, V., 2022. DeFi liquidity management via Optimal Control: Ohm as a case study.

[13] Daian, P., Goldfeder, S., Kell, T., Li, Y., Zhao, X., Bentov, I., ... and Juels, A. (2019). Flash boys 2.0: Frontrunning, transaction reordering, and consensus instability in decentralized exchanges. arXiv preprint arXiv:1904.05234.

[14] Fodra, P. and Labadie, M., 2012. High-frequency market-making with inventory constraints and directional bets. arXiv preprint arXiv:1206.4810.

[15] Guéant, O., Lehalle, C. A., & Tapia, J. (2011). Optimal Portfolio Liquidation with Limit Orders. SIAM Journal on Financial Mathematics, 2(1), 740-764.

[16] Guasoni, P. and Muhle-Karbe, J., 2015. Long horizons, high risk aversion, and endogenous spreads. Mathematical Finance, 25(4), pp.724-753.

[17] Guo, X., De Larrard, A. and Ruan, Z., 2017. Optimal placement in a limit order book: an analytical approach. Mathematics and Financial Economics, 11, pp.189-213.

[18] Huang, R., and Rosenbaum, M. (2015). Optimal market making with asymmetric information and inventory risk. Journal of Applied Probability, 52(1), 1-23.

[19] Kim, J., Lommers, K., Smits, V., and Skidan, B. 2023. HashCurve Protocol White Paper. https://www.hashcurve.com/research-insights/

[20] Kim, J., Lommers, K., Smits, V., and Skidan, B. 2023. Stochastic Automated Market Makers. https://www.hashcurve.com/research-insights/

[21] Kim, J., Lommers, K., Smits, V., and Skidan, B. 2023. Manipulation Resistant NFT Index Oracles. https://www.hashcurve.com/research-insights/

[22] Kominers, S., Teytelboym, A., and Crawford, V. 2017. An invitation to market design. Oxford Review of Economic Policy 33, 4, 541–571.

[23] Law, B. and Viens, F., 2019. Market making under a weakly consistent limit order book model. High Frequency, 2(3-4), pp.215-238.

[24] Law, B. and Viens, F., 2016. Hawkes Processes and Their Applications to High-Frequency Data Modeling. Handbook of High-Frequency Trading and Modeling in Finance, pp.183-219.

[25] Loesch, S., Hindman, N., Richardson, M.B. and Welch, N., 2021. Impermanent Loss in Uniswap v3. arXiv preprint arXiv:2111.09192.

[26] Milionis, J., Moallemi, C.C., Roughgarden, T. and Zhang, A.L., 2022, November. Quantifying loss in automated market makers. In Proceedings of the 2022 ACM CCS Workshop on Decentralized Finance and Security (pp. 71-74).

[27] Park, A., 2021. The conceptual flaws of constant product automated market making. Available at SSRN 3805750.

[28] Robinson, D., and Konstantopoulos, G., 2020. Ethereum is a dark forest. Medium, Aug.

[29] Schöbel, R., and Zhu, J., 1999. Stochastic volatility with an Ornstein–Uhlenbeck process: An extension. Review of Finance, 3(1), 23–46.

[30] samczsun. 2020. So you want to use a price oracle. Personal Blog.

[31] Suresh, A., Latif, M., Shah, N., 2022. So You Still Want to Use a Price Oracle. jumpcrypto.com

[32] Swishchuk, A., Remillard, B., Elliott, R. and Chavez-Casillas, J., 2019. Compound Hawkes processes in limit order books. In Financial Mathematics, Volatility and Covariance Modelling (pp. 191-214). Routledge.

[33] The Block. 2020. DEX to CEX Spot Trading Volume. https://www.theblockcrypto.com/data/decentralized-finance/dex-non-custodial/dex-to-cex-spot-trade-volume

[34] Xu, J., Vavryk, N., Paruch, K. and Cousaert, S. 2021. SoK: Decentralized Exchanges (DEX) with Automated Market Maker (AMM) protocols. arXiv preprint arXiv:2103.12732.

[35] Xu, J., Perez, D., Feng, Y. and Livshits, B., 2023. Auto. gov: Learning-based On-chain Governance for Decentralized Finance (DeFi). arXiv preprint arXiv:2302.09551.