Continuous Double Dutch Auctions

Armann Orn Fridriksson
armann@contango.xyz

Egill Hreinsson
egill@contango.xyz

October 2024

Contents

1 Introduction

The concept of efficient price discovery is a fundamental principle in economic theory. In textbooks, price discovery appears straightforward: An exchange of products takes place at the fair price where the supply and demand curves intersect. In practice, the price discovery process can be interrupted by a multitude of variables. Trader behaviour, market design, and the nature of the asset traded are just the tip of the iceberg.

Large electronic stock- and derivative exchanges like the New York Stock Exchange (NYSE), The Nasdaq Stock Market and The Chicago Mercantile Exchange (CME) are at the center of the global financial system, implementing the predominant market design form, continuous double auctions (CDAs) on limit order books (LOBs)[1].

Electronic Trading & HFT

It is widely accepted that CDAs evolved from the simpler call-auction systems, which became insufficient as trading volumes grew, requiring a more dynamic and efficient matching mechanism. The early success

[1] The limit order book and continuous double auctions may be referred to somewhat interchangeably depending on context.

of CDAs on the NYSE and Nasdaq exchanges accelerated the adoption of CDA across global markets. The LOB further proved its versatility and efficiency in both price discovery and trade execution with the advent of electronic trading.

Furthermore, a convergence between the CDA model, electronic trading, and technological advancements, created fertile ground for a new form of institutional trading, high-frequency trading (HFT). HFT firms can send and receive messages from electronic exchanges with microscopic latencies, allowing them to profit from arbitrage opportunities and market making strategies [1] [2].

Decentralized Finance & Toxic Flow

Technically skilled professionals from traditional finance (TradFi) migrated to the cryptocurrency space as it expanded. The launch of the Ethereum blockchain and the emergence of smart contracts further blurred lines between the two sectors. Consequently, central exchanges in the cryptocurrency space adopted market structures akin to their conventional counterparts, LOBs. Conversely, decentralized exchanges (DEXs) initially developed alternative mechanisms, with automated market makers (AMMs) at the forefront. These designs rely on liquidity pools and formulae to facilitate token swaps, rather than the traditional order books.

However, there has been a recent shift in the DeFi landscape. Emerging decentralized protocols are found employing LOBs.[2] It raises the question whether these structures are appropriate for the unique attributes of cryptocurrencies and blockchain-based assets.

There are several ways in which tools and technologies developed for traditional financial markets do not align with the principles of decentralized finance. On-chain trading often relies on gradual, decentralized price-discovery processes such as AMMs, while identical assets are simultaneosly traded on centralized exchanges with relatively low latency. This leads to exploitable inefficiencies where predatory traders exploit outdated price quotes on DEXs – toxic flow.

DeFi faces several additional challenges. Liquidity in DeFi is often fragmented across multiple chains and protocols, which can amplify the negative impacts of toxic flow. Furthermore, the reliance on on-chain trading and smart contracts introduces scalability limitations; as transaction throughput increases, so do gas fees, reducing overall market efficiency. Another significant challenge is maximal extractable value (MEV), where miners and validators can manipulate transaction ordering within blocks for profit. While MEV-aware solutions, such as Flashbots and time-delay mechanisms, can help reduce these inefficiencies, MEV continues to impact a substantial portion of the DeFi ecosystem, and can exacerbate the risks associated with toxic flow.

Continuous Double Dutch Auctions

This paper introduces a novel market design called Continuous Double Dutch Auctions (CDDAs), along with the concept of the dutch auction order and various functional components aimed at enhancing leveraged decentralized trading. The discussion begins by exploring related work and the complexities of limit order books (LOBs). Next, the CDDA mechanism, dutch auction orders, and multi-stage dutch auction orders are outlined. The paper then delves into the key advantages and vulnerabilities of this design, particularly in relation to existing solutions

[2] Phoenix is a LOB-based DEX, for example.

addressing toxic order flow. Finally, Contango is presented, highlighting the core motivations for adopting CDDAs within the Contango Protocol.

2 Related Implementations

Continuous double dutch auctions are not an entirely original design. They are rather a composition of fundamentals from recent research and novel implementations in the DeFi space. In particular, there are two implementations that bear a strong resemblance to CDDAs and multi-stage dutch auction orders: UniswapX and the CoW Hooks from CoW Swap.

In Section 6 and onwards, a more detailed comparison is drawn between the CDDA system design and these protocols.

2.1 UniswapX

UniswapX is the principal point of comparison for the CDDA and dutch auction orders. This routing protocol, announced in fall 2023, aims to combat issues such as gas costs and MEV by using *fillers* – a combination of market makers, MEV-aware routers, and aggregators amongst others – instead of Uniswap's liquidity pools. The fillers bid for swaps in dutch auctions, competing to offer the best end price for the user, and the winner of the auction ultimately executes the swap. The swaps are implemented as signed intents to trade, and are only settled on-chain when they are filled to avoid transaction costs for unfilled orders [8].

An idealized example of the process is as follows:

1. A trader wants to swap an amount a of token X for token Y. They want to receive at least an ω amount of Y. They set the starting amount of the auction at a reasonable level, generally above the current market price: An α amount of tokens, such that $\alpha > \omega$. They set the auction duration at T time units and apply a linear decay function.
2. At time t_0, the auction proceeds at the initial price α. Presumably, each partaking filler i has an independent valuation of the swap, which includes additional costs, v_i [units of Y].
3. As $t \to T$, the price level, p, decays linearly s.t. $p(t) \to \omega$.
4. In an idealized scenario, a filler will have the optimal valuation $v_j^* \geq v_i \forall i \neq j$, and will "buy" the swap when the price of the swap reaches $p(t) = v_j^*$.
5. If no filler valuates the swap so that $v_i \geq \omega$, the order will expire. Since the order is a signed intent to trade, no costs are incurred.

In theory, this design is similar to CDDAs and dutch auction orders. However, the CDDA implementation addresses two vulnerabilities that Uniswap does not explicitly discuss: the question of signed intents and the potential for gamification via non-meaningful price increments in orders.

CoW Hooks

CoW Swap, a DeFi variant of frequent batch auctions[3], implements so-called *pre- and post-hooks.* The hooks are programmatic operations and invariant checks that enhance flexibility and simplify strategically complicated swaps. A basic application of hooks, for instance, might involve unstaking asset A,

[3] Frequent batch auctions are discussed further in Section 6 and onwards.

exchanging it for asset B, and subsequently staking asset B — all within a single order. While not explicitly atomic, the hooks closely mirror the conceptual entry- and exit-stages in multi-stage dutch auction orders [7].

2.2 Other

1inch is concurrently experimenting with a similar forms of routing in the protocol Fusion. The 1inch Fusion focus is more on minimizing gas fees, while Uniswap's discourse is focused on improving swaps within their own ecosystem. 1inch Fusion is not specifically discussed further in this paper, primarily due to the similarities between the two, and lack of information compared to UniswapX [9].

Dutch auctions are additionally used in some liquidation processes: Instadapp's Fluid, for example, implements a dutch auction variant [10]. Additionally, dutch auctions have been implemented in Non-Fungible Tokens and initial token offerings – these are often variants that imitate dutch auction-based initial public offerings (IPOs) such as the Google IPO.

3 The Limit Order Book

A standard limit order book exchange is structured around a single primitive, the limit order. A limit order comprises four critical parameters:

1. **Asset traded:** The specific asset, which could be a commodity, security, token or derivative.

2. **Quantity or size:** In spot markets, this is referred to as the quantity, while in derivatives markets, it is typically expressed in terms of lot sizes, representing amounts of the underlying asset.[4]

3. **Limit price:** The maximum bid or minimum ask price at which the trader is willing to execute the order, with the intention of being filled at a price equal to or more favorable than the limit price.

4. **Time-in-force:** Refers to the trader's intent regarding the lifecycle of the order. The simplest form defines how long the order remains on the LOB before it is canceled.

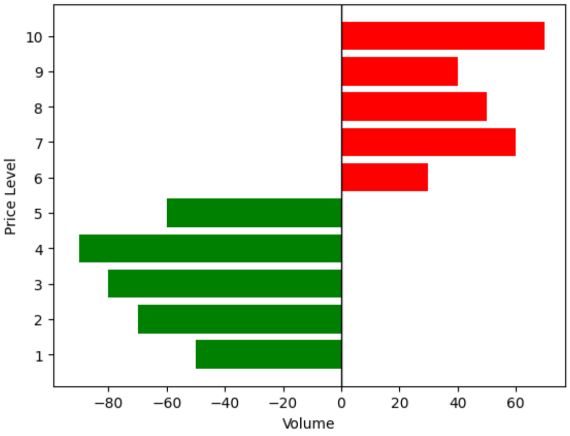

Figure 1: *A sample visualization of a LOB for an asset. Green bars at lower price levels represent the volume of outstanding bids, while red bars at higher price levels indicate the volume of outstanding asks.*

When a limit order cannot be executed immediately, it is placed on the order book. A resting limit order is filled – partially or fully – when a counterparty submits an order with a corresponding limit price on the opposite side of the book. Standard limit order books adhere to a price-time priority rule[5]: if multiple resting orders exist on the same price level,

[4] An example of a lot size is 100 barrels of crude oil.

[5] Some limit order books might implement variations such as pro-rata fills or size-time priority.

those placed earlier will have priority in execution when an opposing order is placed.

If there are orders resting on the book at price levels that are equal to, or more favorable than the specified limit price, taker orders will traverse the order book, beginning by filling resting orders at the optimal price level available. The process – referred to as the aggression phase of the order – continues by executing resting orders at progressively less favorable price levels until either the taker order is fully filled or the liquidity at, or better than, the limit price is depleted.

In LOB markets, price levels are incremented or decremented in fixed intervals, known as ticks. The fundamental premise of market-making in LOBs, before accounting for fees or other benefits, is straightforward. There exist price levels p_a and p_b where the lowest outstanding ask and highest outstanding bid orders are placed, respectively. The difference between those price levels is the spread: $s = p_a - p_b$.[6] A market maker places resting orders on both sides of the order book – a bid at or below p_b, and an ask at or above p_a. If the market maker is filled on both sides of the spread, they realize a profit equal to or greater than s, depending on subsequent market activity.

Making Markets in Limit Order Books

It is important to note the distinction in price improvement for the market makers and the takers. If there are resting orders at a price level more favorable than the taker's limit price, the taker benefits from the price improvement, but pays the price of being the one crossing the spread. The market maker is not afforded the opportunity for price improvement: once a limit order is placed, the maker cannot adjust it and the life-cycle of that order will conclude with either a fill or a cancellation. There are exceptions to this rule, such as cancel-replace orders.

Market makers face several risk factors. A primary risk is that they must be filled on both sides of the book to realize a profit. By definition, for the market maker's bid to be filled, the mid-price of the asset either stays the same or moves downwards, and vice-versa for their asks. Another risk factor is the aforementioned toxic flow. If new information impacts the asset price and a resting order is not canceled in time, it may become stale and susceptible to being filled by a taker who is exploiting the arbitrage opportunity. Market makers are often compensated with fee waivers or other incentives to mitigate their risk.

Order Types

Most LOB exchanges implement multiple order types in addition to the limit order. The majority of these orders are constructed using the aforementioned limit order parameters, and all are executed as limit orders eventually. Recurring examples of such order types include:

1. **Market orders:** Implemented in most public exchanges, a market order executes at the highest bid/lowest ask immediately after it is received. Referring to the aggression phase, these orders are effectively limit orders with an infinite or zero limit price. Some variants offer protection parameters against slippage.[7]

2. **Stop-loss and take-profit orders:** Automat-

[6] Under ideal circumstances, the spread on an LOB is one tick wide.

[7] Slippage refers to the difference between the expected price of a trade and the actual price of execution, and can vary in severity based on market conditions.

ically buying/selling assets when the price of the asset falls or exceeds a predetermined threshold.

3. **Fill-or-kill:** An order that aggresses against the book, and is only filled if the entire order size can be filled at or below the limit price. If the order is not filled in its entirety its canceled. No partial fills and no resting on the order book. Some exchanges also implement a variant called all-or-none which shares the exact same fill logic (no partial fills), but allows for resting on the order book.

4. **Immediate-or-cancel orders:** A type of order that aggresses against the order book and can be partially filled. The quantity that is not matched is removed from the order book immediately.

5. **Good-til-cancel:** A type of order that aggresses against the order book, what can be matched immediately is filled, and the remainder (if there is a remainder) will go rest on the order book until either filled or cancelled. Many exchanges implement variants of this order type where the cancellation is configured to happen at a predetermined time, such as at the end of the current trading session or trading day (good-for-session and good-for-day).

In Section 5, it is demonstrated that replicating the behavior of these orders, even with intricate variations, can be straightforward when using dutch auction orders.

4 Continuous Double Dutch Auctions

The backbone of the continuous double dutch auction (CDDA) is the dutch auction order, as opposed to the limit order of LOBs. Dutch auction orders are further categorized into nominal and tethered orders.

The state of a CDDA at block b[8] is the *order pool*, as opposed to the limit order book, structured as a mixed set of nominal and tethered dutch auction orders:

$$\mathbf{\Phi_b} = \{\phi_i\}_{i=1,2,3,...}$$

where ϕ_i is an order associated with some index i. Orders are removed from the pool when they are matched, canceled, or they expire. For now, the pool is conceptualized entirely on-chain. Refer to Section 8 for a more detailed discussion about on-chain implementations, as opposed to signed intents.

4.1 Nominal dutch auction orders

A nominal dutch auction order is notated

$$\phi_n = (p_\alpha, p_\omega, p_n(b), q, B)$$

where $p_n(b)$ is a weakly monotonic function representing the price of the order, q is the quantity and direction ($q < 0$ for sell orders and vice-versa for buy orders), and B is the duration, in blocks, of the order in the pool. $p_n(b)$[9] is defined on the range $b \in \mathbb{Z}_+ \cap [b_0, b_0 + B]$ where b_0 is the block of order placement, and fulfills the following conditions:

1. For buy orders, $p_n(b + 1) \geq p_n(b), \forall b \in [b_0, b_0 + B[$
2. For sell orders, $p_n(b + 1) \leq p_n(b), \forall b \in [b_0, b_0 + B[$
3. $p_n(b_0) = p_\alpha$ where p_α represents the optimal value the trader wishes to be filled at.
4. $p_n(b_0 + B) = p_\omega$ where p_ω represents the least optimal, yet sufficient price the trader is willing to settle at.

[8] Henceforth, time is measured in blocks to simplify practical aspects of the system description.

[9] The default decay function is a linear decay function.

A nominal dutch auction order therefore has an implied price range $R_p = |p_\omega - p_\alpha|$ and, in absence of fills, terminates after block number $B_\phi = b_0 + B$. A sell-order is effectively a traditional dutch auction and a buy-order is a reverse dutch auction. If $R_\phi = 0$ and B is large, the dutch auction order emulates a limit order.

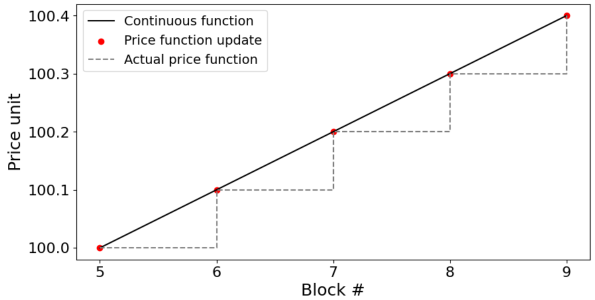

Figure 2: *A linearly increasing price function $p(b) = k \cdot b + a$ for a buy order and its continuous counterpart. The price cannot be approximated continuously. On the Ethereum Virtual Machine, time can pass in blocks on millisecond- or second-scales. On modern day processors, time is incremented at a nanosecond level.*

The price functions are generally notated as a continuous functions, but due to the severe time granularity limitations of the Ethereum Virtual Machine (EVM)[10], they cannot be approximated as such. In practice, it is a step-wise function. A simple visualization of the time granularity effect of the EVM on a linear price function is seen in Figure 2. A further discussion of the implications of the time granularity effect for price granularity is found in Section 4.3.

For future reference, there are additional possibilities in the quantity parameter. Dynamic volume functions offer options such as constantly valued orders or partially entering coveted positions in an illiquid asset market.

4.2 Tethered dutch auction orders

The purpose of the tethered dutch auction order is to facilitate the fair execution of an order at a future point in time. By definition, this is a conditional order. Determining when to begin the aggressive execution phase, and when to conclude it, requires an on-chain reference price that closely tracks the actual market price with acceptable accuracy.

Oracles, which provide these reference prices, typically have a maximum deviation, σ, such that the actual price of the asset at time b, $p_{\text{real}}(b)$ falls within the bounds $p_{\text{real}}(b) = \pi(b)(1 \pm \sigma)$ where $\pi(b)$ is the oracle price at block b. This deviation threshold is implemented since each oracle update incurs a transaction cost, making near-continuous updates expensive and impractical.

A tethered dutch auction order is notated

$$\phi_\pi = (\alpha, \omega, p_\pi(b), \pi(b), q, \lambda)$$

where q holds as the quantity, $\pi(b)$ is an oracle and $[\omega, \alpha]$ are an interval of coefficients that, multiplied by the oracle price, a trader is willing to settle on. $p_\pi(b)$ is, again, a weakly monotonic function that aggresses with regard to the direction of the trade. λ represents the length of the aggression phase, and replaces the order duration from nominal orders. This, in turn, affects the interval on which the price function is defined. If the latest update of the oracle happens on block $b_* > b_0$, where b_0 represents the time of order placement, the tethered price function is defined on the interval $b \in \mathbb{Z}_+ \cap [b_0, b_* + \lambda]$. Furthermore, the

[10] For example, time passes in blocks of up to 15 seconds on the main net.

function fulfills the following conditions:

1. $p_\pi(b+1) \geq p_\pi(b)$ for buy orders and $p_\pi(b+1) \leq p_\pi(b)$ for sell orders, for all $b \in [b_0, b_0 + \lambda[$.
2. $p_\pi(b_0) = \alpha\pi(b_0)$, where b_0 is the placement block.
3. For each point in time, b_*, where the oracle receives an update, the order resets the aggression phase such that $p_\pi(b_*) = \alpha\pi(b_*)$
4. $\alpha\pi(b) \geq p_\pi(b) \geq \omega\pi(b)$ for all $b \in [b_0, b_* + \lambda]$
5. The order terminates when it is filled, cancelled, or finishes its aggression phase without a fill.

The duration of the tethered order, B, therefore functions slightly differently from its nominal counterpart. For the order to expire, the aggression phase must terminate, meaning the oracle price has to keep steady for B blocks. An optional parameter representing blocks until expiration might be passed for flexibility.

A tethered dutch auction order with rationally chosen price coefficients should theoretically be able to rest in the order pool for arbitrary time periods without being sniped – whereas the quote never gets stale – if the oracle resembles the price level on the primary price discovery platform, and is therefore the a line of defence for the trader against toxic flow.

In practice, if the oracle $\pi(b)$ has a maximum deviation of σ[11], the tethered order should fulfill that $-\sigma \leq \omega, \alpha \leq \sigma$. Assuming the actual market price remains within the maximum deviation range, the order will intersect with the real market price before it expires.

For this order type to function correctly – both in terms of ensuring the order is filled and avoiding vulnerability to arbitrage – it is crucial that the real market price[12] consistently falls within the bounds of the oracle price plus or minus the stated deviation. If this condition is not met, arbitrage opportunities may arise, and execution may not materialize. This does not indicate a fundamental flaw in the approach; rather, it underscores the importance of using an accurate oracle.

Numerical Example

In figure 3, there is a visualization of an oracle dutch auction order with the parameters

$$\begin{aligned}\phi_Y = (&\alpha = -10 \text{ bp}, \omega = +10 \text{ bp},\\ &p_\pi = \texttt{lin}, q = +1, \lambda = 8)\end{aligned}$$

The trader's intent is to buy a unit of the asset at a price between -10 bp of the oracle price and 10 bp above the oracle price. The asset is priced at 100 in the beginning, and the oracle is assumed to deviate 10 bp. Assume the oracle is a proxy for the mid-price on the primary price discovery venue for the asset. The duration is 8 blocks and $p_\pi = \texttt{lin}$ indicates that the inverse decay function will be linear according to the order parameters. A linear price function will effectively amount to

$$\begin{aligned}p_\phi(b)|b_* &= \left[\alpha - \frac{\alpha - \omega}{B}(b - b_*)\right]\pi(b_*)\\ &= \left[0.95 - \frac{0.95 - 1.0}{8}(b - b_*)\right]\pi(b_*)\end{aligned}$$

for $b \in [b_0, b_* + B]$, where b_* is the time of the last oracle update. Assume that the price-ticks of the asset are a penny (0.01 units). The chronology of events in the figure:

[11] Maximum deviation is generally measured in basis points.

[12] The real market price is the price at which fillers can execute orders.

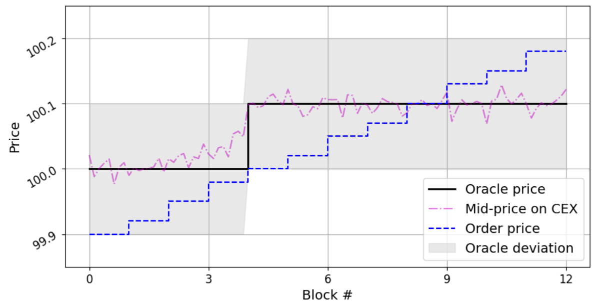

Figure 3: *An example of a tethered dutch auction order. In this case, a trader is looking to buy a lot of some asset, tethering the bid price to an oracle. The bid price aggresses towards the oracle price on $b \in [0, 4[$, when the oracle receives an update at $b_* = 4$. The bid price resets at the same ratio relative to the current oracle price and starts aggressing again. In absence of market makers and oracle updates, the order expires when it finishes the aggression phase at block $b = b_* + \lambda = 12$. Note that the steps are precise to two decimal points, due to the truncation to price-ticks of one penny.*

1. At the placement time, b_0, the oracle shows $\pi(b_0) = 100.0$. The order starts it's aggression phase on the optimal ratio relative to the oracle: $p_\pi(b_0) = 100(1 - 0.001) = 99.9$.

2. On the interval $b \in [0, 4[$, the order aggresses towards the less optimal $\omega\pi(b_0)$. The order will never exceed that threshold.

3. Meanwhile, the mid-price increases until the oracle is updated at $b_* = 4$, such that $\pi(b_*) = 100.1$.

4. The order aggression phase immediately resets at $p_\pi(b_*) = \alpha \cdot \pi(b_*) = 100.1(1 - 0.001) = 99.99$. Since the price-ticks are in pennies, this amounts to $\pi(b_*) = 100$.

5. The order price aggresses towards the less optimal $\omega \cdot \pi(b_*)$. In absence of fills, cancellation or oracle deviations, it terminates at $b_* + B = 12$

For simplification, the price ticks are kept at $\rho = 0.01$, so that all function values fall directly into a tick. However, if the price ticks were less granular, e.g. $\rho = 0.05$, truncation would further obscure the linear dynamic in the aggression phase.

4.3 Time & Price Granularity

The aforementioned challenge of time granularity on the EVM is built into the design at an implementation level. The prices of products traded on the CDDA market must be split into increments. This obstructs gamification via orders with non-meaningful price changes, as discussed in Section 9, and simplifies the relationship between time and price.

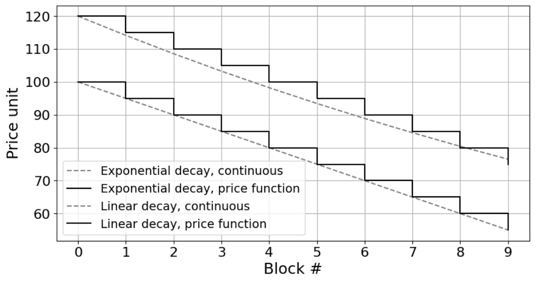

Figure 4: *Two sell orders aggressing in discrete price steps of $\rho = 5$. The price functions are exponential, $p_{exp}(b) = 120 \cdot \exp(-0.05 \cdot b)$, and linear, $p_{lin} = 100 \cdot (1 - 0.05 \cdot b)$. The price granularity, or ticks, are accounted for using ceilings: $\tilde{p}(b) = \lceil \rho^{-1} p(b) \rceil \cdot \rho$ where ρ is the tick size. Note how the values of the exponential function are truncated in the actual price of the order, so the lines do not intersect precisely.*

A suggestion for handing control of the discretization to the trader would be to pass an optional parameter to the order. For example $\delta \in [0, 1]$, could serve as an indicator whether the trader wants to discretizise "forward" (δ or "backward" ($\delta = 1$).

If $p(b)$ is the theoretical price function and $\tilde{p}_b$ is the value the price function takes in practice, this system would calculate the value at block b as

$$\tilde{p}_b = (1 - \delta) \cdot p(b) + \delta \cdot p(b + 1)$$

In Figure 4, a hypothetical example is shown using linear and exponential decay. Both functions would be $\delta = 0$ functions.

4.4 A CDDA Market

A didactic visualization of how the market for an asset might behave over an arbitrary time period is simulated in Figure 5. Assume that the oracle is a proxy for the mid-price at a CEX, which is the primary price discovery venue for the asset. In this specific example, only two orders would be matched in the absence of market makers, where the red (sell) order and the blue (buy) order intersect at $b = 4$. Order B is nominal and, if not matched, will continue to aggress nominally and will not reset at the oracle update. It is unlikely to be filled when compared to the CEX price.

Orders C and D would run into the price-time priority variant applied in CDDAs. To prevent gamification and meaningless price increments in orders, the priority of each price level is given to the oldest order. Therefore, when C and D share a price level at the 8th block, C has execution priority, even though it is aggressing more slowly.

As order E aggresses, the market moves against it. Note that on block 14, as the mid-price diverges from the buy order price, it effectively offers an arbitrage opportunity. The value of these opportunities should theoretically never exceed $2\sigma\pi(b)$ – generally a low value. In a competitive filler environment, the market makers' competition would minimize the cost incurred to the trader.

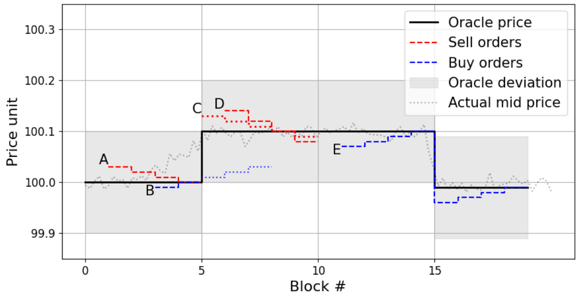

Figure 5: *An example for how a CDDA market for an asset might be visualized. Orders A and B, in absence of any market makers, would be matched on block 4. If order B is nominal, and is not matched with order A, it continues as the dotted lines from $b = 5$, although the mid-price is higher. Order C and order D are both sell orders, but order D aggresses faster. However, should a counterparty fill on the price 100.1 at block $b = 8$, order C will be filled first. Order E would under competitive circumstances be filled before block $b = 15$ judging by the mid-price, but restarts its aggression as the oracle updates.*

In this scenario, the assumption is that the CDDA is not the primary price discovery venue. Under these circumstances, it is highly unlikely that a large portion of transactions are peer-to-peer matches. Instead, fillers are expected to act as the counterparty, with competition between them ensuring close-to-optimal prices.

When the CDDA is a secondary market, the conventional maker-taker roles are effectively reversed: taker orders stand by in the pool, as opposed to maker orders, which rest on the central limit order book. The profitability of microscopic speed advantages is severely limited, especially when front-running protection is applied, as discussed in Section 9. Only when two makers value the settlement of the asset at exactly the same price in exactly the same block will speed advantages matter.

The CDDA could theoretically become a primary price discovery venue for an asset. Under those conditions, peer-to-peer matching plays a much more important role. In any market, the price of an asset is determined by genuine buyers and sellers, and the role of the market maker on the CDDA would be diminished to a degree. In a scenario where price discovery lies somewhere between the CDDA and other exchanges, there is no obvious reason as to why the CDDA would underperform compared to alternative designs.

5 Multi-Stage Dutch Auction Orders

Multi-stage dutch auction orders (multi-stage dutch auction orders) are order-types that enable complex conditional transactions in a CDDA market. The entry- and exit-stages of an order are defined as ordered mixed sets of invariants that must be fulfilled and operations that must be executed successfully, for an order to be settled. Any actions and invariants that can be embedded in smart-contracts are fair game: Oracle checks, flash-loans, staking and unstaking are just a few of the possibilities. Implementing chains of multi-stage dutch auction orders using the entry and exit stages vastly expands the potential applications.

The activation indicator is a crucial component of these orders. The activation indicator is an invariant in the that represents whether an order is executable. It could, for example, adhere to time signals, price oracles, or any blockchain compatible function. This is immensely important for flexibility in more advanced trading strategies.

However, allowing for arbitrary activation indicators, which are not possible to verify on-chain, is the key to unlocking a vast universe of possible use-cases. Leveraging the latest advancements in zero-knowledge cryptography enables this. Specifically, zk-SNARKs can be utilized to validate invariants at constant time-complexity during settlement. Verifying an invariant will carry the same transaction cost, regardless of the complexity of the invariant. The next hard fork of Ethereum, Pectra, is scheduled to go live in Q1 of 2025. Pectra adds new pre-compile contracts to the EVM that make verification of zero-knowledge proofs available at a feasible transaction cost.

Order types such as market orders, stop-loss/take-profit, all-or-none, immediate-or-cancel and fill-or-kill,[13] are emulated easily using multi-stage dutch auction orders. A market order could be a dutch auction order with a starting price based on a realistic signal, for example some price oracle for the asset, and should be filled by the lowest (highest) ask (bid) in the pool, given that they are priced reasonably. Stop-loss and take-profit orders could be implemented by pseudo-limit orders, using the activation indicator at the entry stage. The all-or-none, immediate-or-cancel and fill-or-kill orders could all be solved by invariant checks on the entry and exit

[13] Refer to Section 3 for a description of these order types.

stages.

In the following sections, a few examples of use cases for multi-stage dutch auction orders are listed.

Dynamic Stop-Loss Orders

A bullish trader wants to buy q units of asset X. A unit of X is trading at roughly 100 quote units [qu], and X and its derivatives can be quite volatile on markets. He is not sensitive to the in-price and sets a dutch auction order $\phi_X = (p_\alpha = 99, p_\omega = 101, q, B)$, where B is some arbitrary unit of time.

He wants to sell for a loss if the price X falls to 95 [qu], however, preferably not in a temporary bout of volatility – for example, when the mid-price of X has not exceeded 95 [qu] for $a \in \mathbb{Z}_+$ time units. If he receives a proxy signal for the mid-price of X through some oracle $\pi(b)$, he can set up a pseudo stop-loss order $\phi_{X,s} = (p_\omega = p_\alpha = 95, -q, B_s = z)$ where z is a large integer and the activation indicator is $c(b) = \mathbf{I}_{\pi(\beta)<95 \forall \beta \in [b-a,b] \cap F(b)}$, where $\mathbf{I}$ is the indicator function and F indicates that the original buy order has been filled at some point before b.

Time-Weighted Average Price Orders

A patient, bullish trader notes that she wants to buy q units of X before midnight. However, she notes that market volatility is rather high, and opts to spread her volatility risk over the rest of the day. The time is 19:30, and her strategy is to trade in five settlements (at 20:00, 21:00, ...). She can split the volume into 5 batches of $q/5$, and set the activation indicator to go off at 20.00. She can chain 4 of these in the post-stage for 21:00, 22:00,... Assuming she is filled on roughly the oracle price at each step of the TWAP, her effective fill price on the q order is an approximate average of the price over the four hours in question.

This process can automated for numerous small portions over longer periods, so institutional traders can alleviate potential market impact in large trades.

Parallel Orders

Flow trading is proposed by Eric Budish et. al. [4] and one of the fundamental components of the system is an order that comprises of weights, resembling a portfolio. It is argued that portfolio-like trading is more in line with the complex strategies applied in modern financial markets.

Although there are no weight-like parameters in the multi-stage dutch auction orders, a party can deliver a batch of chained multi-stage dutch auction orders with coinciding invariants at the exit stage as a facsimile for a portfolio-, or ETF-like order.

Collateral Auctions

Collateral auctions – or similar mechanisms – are necessary for lending protocols to prevent the accumulation of bad debt, and are conducted using various auction structures through-out the DeFi space. Using dutch auction orders should theoretically lead to optimal value retrieval, as opposed to using ratios or nominal asks for the same collateral. Implementing collateral auctions could possibly involve tethered dutch auction orders based on predetermined oracles by the exchange, effectively rerouting these auctions to the DEX itself.

6 Key Improvements

Despite advancements in technology that have rendered early hardware and software constraints largely obsolete, market designers remain focused on engineering new order types to mitigate the deficiencies of a system that inherently reduces flexibility. CDDAs and multi-stage range orders offer numerous advantages over limit order books and alternative DeFi designs:

1. **Toxic Flow:** Tethered dutch auction orders near-eliminate toxic flow if the oracle provided is accurate enough. The theoretical maximum value of arbitrage opportunities is low and market makers compete to decrease their duration.

2. **Price Priority:** Generally, price is prioritized over timing. Time-priority is greatly diminished, although there are queues on price-levels to improve fairness.

3. **Time Flexibility:** Although bound within the constraints of the EVM time granularity and quasi-subpenny rules, CDDAs are a step in the right direction regarding both time- and price-continuity. There is always a lingering potential for faster computation and CDDAs can adapt to faster- and slower blockchains.

4. **Complex Trading:** The multi-stage range orders offers interconnected, multi-directional and multi-asset trades that can facilitate challenging trading strategies. Limit order books constrain the same strategies due to their prioritization of time and discrete nature.

5. **MEV Accounting Shift:** The risk of miner/maximal extractable value is intrinsic to blockchains. Miners or validators can, for example, exploit transaction sequences to front-run, back-run and/or execute sandwich attacks. Gas fees are another consideration, and dynamic according to blockchain throughput, exacerbating price volatility and the cost of trading. Continuous double dutch auctions shift the responsibility of MEV protection and transaction fees from the user to the market makers, who are incentivized to minimize the costs of trading to successfully capture orders to settle. The traders additionally benefit from a more transparent overview of the total cost of trading.

7 Alternative Solutions to Toxic Flow

The issue of toxic flow in modern markets arises from information and latency asymmetries. In order book systems, new information can render resting orders outdated or "stale," triggering predatory HFT strategies that exploit speed advantages to execute trades before these stale orders are canceled. In DeFi market designs like AMMs, toxic flow contributes to the broader challenge of MEV.

Eric Budish et al. [3] advocate for frequent batch auctions. They show that on HFT time-scales, price correlations between closely related products break down, creating arbitrage opportunities. Crucially, the nature of these opportunities remains constant over time, but the windows to exploit them shrink as latency improves.

Baron et al. [11] differentiate between passive HFTs, which provide liquidity, and aggressive HFTs, which exploit arbitrage by sniping stale orders. They estimate that 45% of aggressive HFT profits come at the expense of passive market makers, with up to 20% of total trading volume involving stale quote sniping. Anirban Banerjee and Prince Roy [12] further show that some regulatory structures incentivize HFTs to take liquidity rather than provide it.

These studies highlight a winner-takes-all environment, where microscopic speed advantages are decisive, while the often-cited benefits of HFT—such as increased liquidity and tighter spreads—have not meaningfully improved conditions for retail traders.

Toxic flow, particularly in the context of HFT, has prompted a variety of responses over time, which are outlined in this section. Several of these topics are explored in greater depth in Budish et al.'s [3] arguments for frequent batch auctions (FBAs).

7.1 Tobin Taxes and MTRs

Tobin taxes were proposed in 1978, long before the conception of HFT. A Tobin tax imposes a fee to each transaction, and was initially suggested to reduce what was regarded as excessive speculation and volatility in foreign exchange markets. In the 21st century, this tax resurfaced as a viable option to curtail the advantages of HFT, as firms that trade in higher volumes would incur greater costs.

This approach does not directly address the challenge of stale-quote sniping, and presents an obvious trade-off, negatively impacting the retail traders it is meant to protect.

Imposing thresholds on message-to-trade ratios has proven just as inadequate. As Baruch and Glosten [6] demonstrate, flickering orders – rapid order cancellations and replacements – are an equilibrium outcome in limit order books. High message-to-trade ratios are a byproduct of the system design, not a deliberate tactic employed by HFT firms. Meanwhile, requiring minimum resting periods for orders are a perverse incentive: they more-or-less ensure that traders are not able to cancel stale quotes before arbitrageurs snipe them [3].

7.2 Message Delays

A notable response to toxic flow came from Brad Katsuyama, whose efforts were popularized by Michael Lewis in his non-fiction *Flash boys* from 2014. Disillusioned by consistently losing to predatory and minimally unregulated HFT firms, Katsuyama cofounded The Investor's Exchange (IEX). IEX was built on the principle of reducing the advantages of marginal speed gains, to the benefit of retail traders.

The backbone of the IEX defense against HFT was the implementation of a deterministic message delay to immediately executable orders, while cancel and cancel-replace orders were routed straight to the matching engine, bids and asks were sent through a copper wire that effectively introduced a fixed 350 microsecond delay. This gives liquidity providers a head start over stale-quote snipers in a race to react.

Several other exchanges have since adopted message delay variants for similar purposes, using two primary approaches: Constant-time delays and randomized delays. Randomized message delays, such as those introduced by ParFX, are generally implemented for all messages – in which case, they do not address stale quote sniping in particular, and rather add a stochastic element to the result of the competition among HFT firms. Introducing fixed delays to immediately executable orders, that exceed the speed advantage HFTs have over retail traders, has proven to be an effective strategy [3].

The asymmetric message delay does not address the race to the top of the book. Such a race exists when the minimum price increment[14] is large, compared to what it would be in the absence of such a constraint.

This race has been demonstrated to occur regularly [14]. Particularly, it is in US regulated markets due to

[14] The minimum price increment is generally referred to as the tick size.

the uniform one-cent tick size imposed by the SEC.[15] The EU regulation governing tick sizes, under MiFID II[16] and MiFIR, is more dynamic than the American subpenny rule, as it considers a broader set of factors when determining minimum tick sizes. In this context, trading firms strictly prefer acting as liquidity providers over engaging in stale-quote sniping.

In equilibrium, a price movement triggers two races: one to snipe stale quotes, and another to secure a top position in the order book to provide liquidity at the new price level. While constant message delays primarily address the first race, while CDDAs and the FBAs described below acknowledge both of them to some degree.

7.3 FBAs & Hyperliquid

Frequent Batch Auctions (FBAs) address toxic flow by batching orders over a set period and executing them simultaneously at a uniform clearing price. Microscopic speed advantages become irrelevant, and responsive liquidity providers can almost always cancel outdated quotes in time. During sudden market shifts, stale quotes that remain on the book are likely to be cleared at a more favorable price as the market responds efficiently.

Hyperliquid offers a solution that lies somewhat ambiguously between frequent batch auctions (FBAs) and fixed message delays. On their proprietary blockchain, cancel and post-only orders are prioritized over immediately executable orders, effectively processing transactions in blocks – or batches. This introduces a "delay" for immediate orders, although the time is neither constant nor entirely random. While the order of transactions remains intact after cancellations, it does not fully eliminate the race to exploit stale quotes if they aren't canceled in time.

In DeFi, participation poses challenges for implementing FBAs. For example, CoW Swap uses frequent batch auctions for peer-to-peer matching, but most trading volume is handled by "solvers"– a market-making system similar to the makers proposed in CDDAs. In these cases, dutch auction mechanisms, driven by competition on price and time, should lead to more efficient outcomes.

7.4 DeFi Space

The automated market maker (AMM) is the cornerstone of decentralized exchanges (DEXs). AMM platforms, such as Uniswap, rely on liquidity pools and predetermined pricing formulas[17] for price discovery. Ideally, AMMs are a fair representation of supply and demand. However, when they are not the primary venue for price discovery of the two assets, toxic flow becomes intrinsic to their structure: price updates unfold gradually based on trades within the pool, giving arbitrageurs frequent opportunities, especially when nearly instantaneous price corrections for the same assets occur on more sophisticated exchanges.

One proposed solution to mitigate constant price slippage is virtual automated market makers (vAMMs). These platforms emulate constant product AMMs utilizing virtual assets, removing the need for liquidity providers. Yet, toxic flow remains inherent to these exchanges. While vAMMs resolve some

[15] Rule 612, "the subpenny rule", is readable here: https://www.law.cornell.edu/cfr/text/17/242.612

[16] Specifically, article 49: https://www.esma.europa.eu/publications-and-data/interactive-single-rulebook/mifid-ii/article-49-tick-sizes.

[17] The constant product market maker, $xy = k$, is a standard example, where x and y are the quantities of the two tokens in the liquidity pool, and k is a constant.

of the limitations of traditional AMMs, they introduce new challenges in maintaining price alignment with the broader market [13].

In general, price discovery for reasonably liquid blockchain assets does not take place on DEXs. Consequently, on-chain limit order books and AMMs consistently suffer from toxic flow. Most existing solutions for improving prices either circumvent the problem by minimizing losses or rely on off-chain liquidity providers—both approaches are found in certain intent-based trading solutions.

8 Signed Intents

The starkest distinction between UniswapX and CDDAs lies in the order implementation strategies. While UniswapX's dutch orders are executed as signed intents, CDDAs are designed for fully on-chain orders.

One key advantage of signed intents is the absence of transaction fees for unfilled orders. However, the subsequent section will delve into the potential weaknesses of signed intents, particularly in areas such as message spamming, order cancellations, and order amendments.

This section will not discuss the issue of non-meaningful incremental orders – although in some part related to message spamming – whereas this challenge is broader in nature and also applies to CDDAs. For further discussion, refer to Section 9.

8.1 Message Spamming

In the absence of order costs or protective mechanisms, there is little to deter market participants from flooding the exchange with excessive order messages. This behavior, often referred to as *message spamming*, can be driven by various motives, none of which serve the broader market interest.

For instance, participants may engage in spam to manipulate market sentiment, creating the illusion of liquidity or demand where little actually exists – spoofing. Another common tactic is to overload the exchange infrastructure, disrupting operations for competitive advantage or to exploit latency for high-frequency trading strategies.

There are two logical solutions to this challenge: fees and trust.

Fees

Introducing fees is a practical solution to deter excessive order posting. Dynamic fee structures, which adjust based on overall message posting demand, offer a more viable approach than pro-rata fees based solely on trading volume. The latter fails to address the issue of flooding the exchange with numerous small-quantity orders.

Although dynamic fees are discriminatory in favour of wealthier participants, this model has proven practical in low-trust environments. It mirrors the approach used by blockchains, where such mechanisms promote fairness by limiting resource abuse while maintaining system integrity.

However, using an off-chain relay network with fees to manage blockchain assets is questionable, as the blockchain itself already employs this model. If the relay system were attacked, a fallback to the underlying blockchain is the viable strategy and further obscures the purpose of the off-chain system.

Trust

The second option involves gatekeeping access using a trust mechanism. These mechanisms must inherently be able to meaningfully punish abuse and prevent recurrence.

Centralized trading venues use a combination of fees and trust to protect their infrastructure, with all network sessions being authenticated and stateful. If a user violates rate limits, the session is terminated, and reconnection attempts are screened by a load balancer. If deemed malicious, the user is blocked from reconnecting. Attacks require prior authentication, allowing the exchange to trace and cut off the offender at the authentication level, which also represents the only real centralized point of failure in terms of message spamming.

While an on-chain exchange could mimic this CEX model, it contradicts the core principles of decentralized finance. A private or semi-private execution layer for certain assets might be a fitting use case, but not for a public DEX.

8.2 Canceling & Amending Orders

Order cancellations and amendments in off-chain relay networks, such as the one proposed by UniswapX, face limitations due to their non-stateful nature. Off-chain networks cannot natively support cancellations or modifications of orders once posted. A cryptographically signed order can only be invalidated by taking an action on-chain, such as submitting a transaction with the same nonce, which effectively cancels the original order.

This poses a problem for resting limit orders, where users may want to modify their orders without incurring the cost of on-chain transactions. While price improvements can be handled off-chain by signing a new order with the same nonce – since the more attractively priced order will always be filled first– price reductions require on-chain action. This is paradoxical: The off-chain system requires on-chain transactions.

For example, for a limit order to buy a token at a price p_0 with the nonce X, one can only amend the order off-chain by signing another order, with the same nonce X, using a price $p_* > p_0$. This process functions based on economic principles rather than mechanic logic – that the order with the more favorable price will be executed first, and invalidate the nonce of the previous order.

Theoretically, this approach is similar to how transaction cancellations work on Ethereum. In order to cancel a transaction, a new transaction is submitted with a higher priority fee, and the economic incentive will almost surely result in the latter transaction's inclusion ahead of the former in a block.

If the trader rather wants to amend a resting order where the price is lower, $p_* < p_0$, the same economic assumptions will not invalidate the previous order. The only reliable way to amend the order (or completely cancel it) requires on-chain action. The off-chain relay network therefore only allows you to post an order - amending it or cancelling will generally rely on the blockchain.

The primary reason for using an off-chain network is to improve user experience by enabling ERC20 swaps without requiring native tokens for transaction fees. This is particularly important as the number of EVM blockchains increases. However, users my still need to cancel or amend resting orders on-chain, which requires the native token.

A potential workaround is using the Permit2 deadline feature to automatically cancel and re-submit orders periodically, which could simulate free order cancellations. However, since each order requires signing with a private key, this method can't be easily delegated without compromising token security.

The dutch auction orders of the CDDA model are not designed to be canceled – they are aggressive orders, intended to be filled during their lifecycle. If the dutch auction orders are not filled, they are terminated. For the purposes of flexibility, however, a combination of on-chain orders and signed intents is a realistic possibility, and the problems associated with it are included as a future reference.

9 Vulnerabilities

As of late, there has been a lot of discussion and investment in so-called *intent based protocols* in the DeFi space. This term has been used as an umbrella term for anything that involves asynchronous execution of trades. Similar to the traditional relationship between a client and a broker, a trader states their investment intent, and the broker deals with executing that intent on their behalf.

Several of the proposed designs are intrinsically flawed. Some obvious flaws have already been discussed in the previous section: trust- and stateful systems and the potential points of attack. Request-for-quote (RFQ) based intent systems are an example of trustless failure, whereas RFQ is exploitable when no counter-party trust exists.

Out of all the intents based platforms already built or are being built, we've found none that address or even identify the following concerns:

Minimum Price Increments

A market design that minimizes the spread is not always in the best interest of retail traders. There's a trade-off between having a small spread and maintaining liquidity. The spread reflects the cost of liquidity, paid by traders who seek immediate execution. A wider spread typically leads to deeper liquidity but lower trading demand, while a tighter spread results in higher demand but shallower liquidity.

Minimum tick sizes as outlined in Section 7.2, enforced by all limit order books, regulate the smallest price increment on an exchange. This rule manages the balance between trading costs and liquidity depth, often mandated by regulators, but also self-imposed by exchanges to meet liquidity objectives. Minimum tick sizes also prevent high-frequency traders from "gaming" the system by marginally improving prices to gain order priority, which can harm fair execution for retail and institutional traders.

In DeFi, existing implementations of limit and dutch auction orders, like UniswapX and Paraswap, do not enforce a minimum price increment, with price granularity limited only by token decimals.[18] This allows traders to strategically place orders slightly ahead of larger ones to gain priority.

Price-Time Priority

Modeling a market as an unordered pool of orders has certain advantages, such as embedding gas costs directly into the fill price, simplifying the complexity of how traders pay for order settlements. Settlement logic must also apply price-time priority:

1. **Liquidity Provision:** Time priority incentivizes market participants to submit limit orders early, thereby enhancing market liquidity. Without time priority, participants would have less motivation to submit orders in advance, as newer orders could easily displace earlier ones. By prioritizing the earliest orders at the same price level, the system rewards those providing

[18] To the best of our knowledge, acquired in whitepapers and articles.

liquidity, which contributes to price stability and reduced volatility.

2. **Manipulation Prevention:** In the absence of time priority, larger or faster traders could exploit the system by frequently modifying or canceling orders at high speeds, a practice known as "spoofing" or "layering." These orders are not intended to be executed but serve to create misleading signals regarding market demand. Time priority mitigates this by ensuring that large spoofing orders remain at the back of the price queue, making such manipulative tactics more identifiable and easier to disregard.

3. **Price Discovery:** Time priority plays a crucial role in promoting efficient price discovery. When orders at the same price level are executed in the order they arrive, the flow of orders more accurately reflects genuine supply and demand dynamics over time. Without time priority, assessing true market interest at particular price levels would be more difficult, as newer orders could continually take precedence.

4. **Trust in Market Fairness:** Finally, the perception of fairness is essential for the integrity of any market. Time priority fosters trust by ensuring that participants who place orders first at a given price receive priority over later entrants. If this principle were compromised, confidence in the fairness and transparency of the market would diminish, potentially leading to reduced participation and liquidity. Additionally, filling larger orders placed later over smaller, earlier orders undermines trust and opens the door to "bribing" makers to prioritize certain orders. This practice could effectively circumvent the concept of minimum price increments, as participants could place orders between discrete price levels, weakening market structure integrity.

While these issues may not present an immediate risk in the early stages of product development, overlooking them in the long term is likely to result in problems as the market matures. Initial design flaws may go unnoticed with limited user engagement, but swaps are an already validated use case. The market design should be robust enough to operate as a standalone, mature structure rather than relying solely on functioning as a proxy for centralized exchanges (CEX). As the market grows and becomes a primary venue for trading certain pairs, the risk of exploitation will inevitably emerge if time priority and other key principles are not properly enforced.

10 Contango V3

DeFi money markets provide users the ability to borrow against their crypto assets, maintaining price exposure while accessing liquidity. Several traders loop their assets by borrowing, buying more of the same asset, and re-depositing it as collateral. This strategy, though repetitive and complex, mimics traditional margin loans but with cheaper and more stable funding compared to perpetual futures contracts [15]. The process, often referred to as “looping,” lets traders maximize exposure to a particular asset while earning yield from the lending market.

About Contango

Contango offers one-click leveraged trading on decentralized money markets. The process of looping is simplified by securing flash loans and executing only one swap for the full trade in one go. Contango handles everything under the hood, including sourcing swaps, simulating transactions, and submitting them to smart contracts for execution, providing a seamless and efficient experience for users while eliminating the need for recursive actions.

We propose extending what is effectively the order mechanism offered by UniswapX, as described in Section 2, to include leverage. An order type that would allow for arbitrary code execution during the lifecycle of the settlement phase. A pre-hook could pull user funds and initiate a flash loan, combining both as the swap input. Once a swap is executed, tokens received would be deposited into a lending market, allowing the user to borrow against the deposit and repay the flash loan. This structure would simplify the leverage process further, offering an intuitive and efficient way for users to increase their market exposure while ensuring all trades are collateralized.

Improving the Design

Addressing the issues discussed in Section 9, we propose an implementation of intent order types where each swap combination of tokens A and B has a defined minimum price increment. For dutch auction orders, the price will aggress from its starting point toward the end price[19], incrementing toward the end price based on the number of blocks. The order should specify the number of blocks that must pass before incrementing by one price level.

Selecting an appropriate aggression speed depends on factors such as market maturity and the blockchain used, and should therefore remain configurable. Orders can only be settled at defined price intervals, which vary based on the assets being traded—similar to all traditional limit order books. This would eliminate attempts to gain execution priority by marginally improving the price.

Additionally, we propose a system that strictly enforces time priority for both regular limit orders and dutch auction orders. Although this complicates implementation, particularly with respect to managing settlement costs, we believe it is essential to ensuring a fair and transparent market. The strict enforcement of minimum ticks and price-time priority—standard practice across all limit order books—along with the other arguments presented in this paper, underscores their critical importance.

Success So Far

Contango's success so far has been largely driven by its offering of leveraged positions on yield-bearing ETH derivative pairs, such as wstETH/ETH. These products enable traders to earn yield with minimal price exposure. Price discovery for these derivative token pairs primarily occurs on decentralized exchanges (DEXs), rather than on centralized platforms like Binance. It is reasonable to speculate that the DEXs and aggregators used by Contango ensure near-perfect taker prices for these derivatives. In contrast, products traded mainly on centralized exchanges are lacking at the price level, likely due to the toxic flow issues discussed in this paper.

Building on that argument, for the volume of directional products[20] traded on Contango to match or surpass the volume of derivative token pairs, the fill prices must improve. The current design, which relies on taker orders on DEXs, places directional product prices at a significant disadvantage. This is be-

[19] The client determines the price range.

[20] Directional token pairs such as ETH/USDC are primarily traded on centralized exchanges (CEXs).

cause price discovery inefficiencies become more pronounced in speculative trading, where net profits are determined by fill prices. While the low and stable funding costs of cPerps help offset fill costs in long- and mid-term trades, Contango is effectively priced out of the market for short-term leveraged trades.

A high-level diagram of the new design is shown in Figure 6. With the advantages of the CDDA market design, these products not only make short-term trading on DeFi markets more attractive, but also offer long-term positions that capture the best of both worlds:

1. More stable and lower funding rates.

2. Entry and exit prices comparable to those on centralized exchanges.

Our research indicates that DeFi money markets are already a significant source of margin loans for directional trading. On-chain data reveals single-wallet ETH/USD positions reaching several hundreds of millions. A common strategy involves using Aave as the lending platform and Binance as the spot venue. Traders typically build up these positions recursively—borrowing on Aave, depositing in Binance, buying ETH on the spot market, withdrawing ETH from Binance, depositing it back into Aave, and repeating the process.

While this strategy might be feasible for large traders, it is a slow and tedious process. Additionally, managing the accounting – especially when using multiple spot venues – demands significant effort or investment in specialized tracking and accounting tools. Executing the strategy optimally also requires careful consideration of how and where to route the spot orders. Optimizing order routing would likely be more efficiently handled by a network of specialized fillers, each competing with each other.

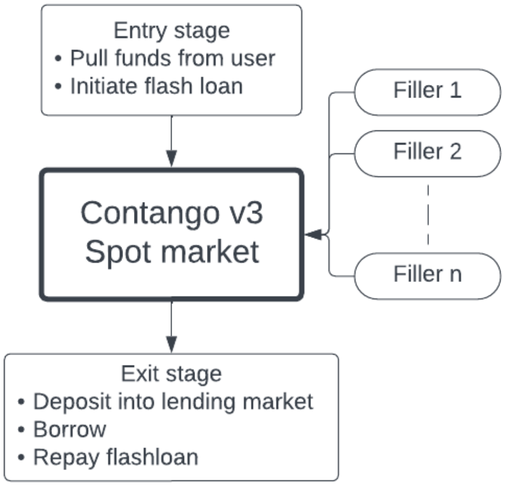

Figure 6: *The design for a new cPerp involves a Multi-Stage dutch auction order (multi-stage dutch auction orders) that emulates the looping strategy. It works by taking a flash loan at the entry stage, executing a swap on a CDDA market, and repaying the flash loan at the exit stage. The CDDA architecture, along with specialized fillers, optimizes the order routing for the best possible fill price, compensating for previous inefficiencies in price discovery, specifically in short-term directional trades.*

Roadmap

We believe that Contango is well positioned to become the leading DeFi brokerage business. A typical brokerage serves two main functions:

1. Providing traders with access to margin loans.

2. Order routing, or trade execution

To date, Contango has integrated with 17 different DeFi money markets to facilitate margin loans. Each market offers a unique combination of collateral and debt options, enabling us to offer a broad range of trading pairs (currently over 290). For trade execution, we use DEX aggregators to source swaps for our traders, pulling from over 20 different DEX aggregators to ensure the best possible swap price. With Contango v3, we aim to enhance this by adopting an intent-based order routing system, which we believe will make us more competitive, particularly in heavily traded blue-chip asset pairs. We combine money market integration and order routing with an advanced trading UI. While a good user interface is crucial for a quality brokerage, it must be paired with competitive trade execution, which this new design intends to address.

It's also important to note that Contango v3 has use cases beyond our current product offering. For instance, the programmatic dutch auction order type is ideal for liquidating positions on money markets. DeFi money markets have steadily increased their loan-to-value (LTV) ratios, making Contango more competitive in leveraged trading. A more robust and efficient spot market, as proposed here, will enable even higher LTV ratios. Currently, LTV ratios in DeFi are lower than those on centralized exchanges primarily due to liquidity limitations in the DeFi spot market. Platforms like Contango v3 will help increase these ratios significantly, making DeFi more competitive.

References

[1] Anna Calamia, *Market Microstructure: Theory and Empirics*, Cambridge University Press, 1999.

[2] Harris, Lawrence. *Trading and Exchanges: Market Microstructure for Practitioners*, Oxford University Press, 2003.

[3] Eric Budish, Peter Cramton, and John Shim, *The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response*, The Quarterly Journal of Economics, Volume 130, Issue 4, Pages 1547–1621, Published: 23 July 2015. DOI: 10.1093/qje/qjv027.

[4] Eric Budish, Peter Cramton, Albert S. Kyle, Jeongmin Lee & David Malec. *Flow Trading.* 2022. Preliminary version retrieved at `https://www.econtribute.de/RePEc/ajk/ajkdps/ECONtribute_146_2022.pdf`.

[5] Ivan Indriawan, Roberto Pascual, and Andriy Shkilko, *On the Effects of Continuous Trading*, March 3, 2021.

[6] Baruch, Shmuel and Glosten, Lawrence R., Fleeting Orders (June 11, 2013). Columbia Business School Research Paper No. 13-43, Available at SSRN: `https://ssrn.com/abstract=2278457orhttp://dx.doi.org/10.2139/ssrn.2278457`

[7] CoW Protocol, *CoW Protocol Documentation*, 2024. Available at: `https://docs.cow.fi/cow-protocol/`. Accessed on August 26, 2024.

[8] Hayden Adams, Noah Zinsmeister, Mark Toda, Emily Williams, Xin Wan, Matteo Leibowitz, Will Pote, Allen Lin, Eric Zhong, Zhiyuan Yang, Riley Campbell, Alex Karys, and Dan Robinson, *UniswapX*, July 2023. `https://people.eecs.berkeley.edu/~ksk/files/GDA.pdf`.

[9] 1inch, *1inch Fusion 2.0 Revolutionizes Swap Efficiency for Users*, 1inch Blog, 2023. `https://tinyurl.com/cfuwk3d2`.

[10] Jain, Samyak. *Introducing Fluid!*. Instadapp Blog, 2023. `https://blog.instadapp.io/fluid/`.

[11] Matthew Baron, Jonathan Brogaard, and Andrei Kirilenko, *Risk and Return in High Frequency Trading*, 2014.

[12] Anirban Banerjee and Prince Roy, *High-frequency traders' evolving role as market makers*, Indian Institute of Management Ahmedabad, India, and State University of New York (SUNY) at Buffalo, United States of America. 2023. `https://tinyurl.com/yn84rsa7`.

[13] HyperLiquid. *The Hyperoptimized Order Book.* Available at: `https://newsletter.asxn.xyz/p/hyperliquid-the-hyperoptimized-order`, 2023.

[14] Chengxi Yao, Mao Ye, *Tick Size Constraints , High-Frequency Trading , and Liquidity*

[15] Lukasz Szpruch, Jiahua Xu, Marc Sabate-Vidales, Kamel Aouane, *Leveraged Trading Via Lending Platforms*, 2024. `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4713126`