

CARBON

a decentralized protocol for asymmetric liquidity and trading

Litepaper

November 2022 | Working Draft v1.1

Overview	3
Background	4
Asymmetric Liquidity	6
Adjustable Bonding Curves	8
Strategies	9
Asymmetric Liquidity Pools	10
MEV Resistance	10
Carbon Implementation	11
Protocol Fees	12
Use Cases & Simulations	12
Limit Order	12
Buy Low, Sell High Strategy	13
Average-In Order	13
Token Distribution	13
Conclusion	13
APPENDIX	15
Glossary	15
Resources	16

Overview

In this paper, we introduce the concept of Asymmetric Liquidity, a new form of on-chain liquidity that enables trading and active market-making strategies defined by multiple bonding curves, simultaneously. Instead of users providing liquidity to a single curve that trades symmetrically in both directions, users provide liquidity to two curves that each trade in one direction. In this design, the buying and selling of an asset are governed by separate, user-defined curves, giving users greater control to express their trading preferences.

The one-directional nature of Asymmetric Liquidity is desirable for users who wish to commit to a premeditated trading strategy. Each strategy is composed of on-chain limit orders and range orders that execute continuously and irreversibly with no dependence on oracles or keepers. A user can, for example, deploy a strategy where one order buys ETH between 1200 and 1300 USDC and the other order sells ETH between 1500 and 1600 USDC. ETH accumulated in the first order becomes available to sell for USDC as prices cross into the range defined in the second order.

Each individual order in the system has a corresponding, uniquely defined bonding curve. An order's underlying curve is determined by parameters defining the curve's shape (e.g., constant sum vs. constant product) and the price range where liquidity is concentrated. A curve's parameters may be updated directly without closing and recreating the liquidity position, allowing for gas-efficient changes to order conditions. Aggregated together, strategies and their composite orders create on-chain liquidity for a given token pair, with a routing engine determining the optimal path and liquidity utilization for trades against the network. Due to the asymmetry of the system, trading is resistant to sandwich attacks, the most common form of Maximal Extractable Value ("MEV").

Herein, we present the high-level mechanics of Asymmetric Liquidity and propose its first implementation, Carbon. Carbon unleashes trading and active market-making strategies in DeFi that have historically only been accessible in CeFi. These capabilities aim to bridge CeFi liquidity into DeFi, and position DeFi as a true challenger to traditional centralized finance.

Carbon Features:

- **Asymmetric and Irreversible:** individual user strategies may be composed of independent buy and sell orders which trade in a single direction and are therefore irreversible on execution.
- **Concentrated and Adjustable:** order conditions are pre-defined using specific concentrated ranges, and may be updated on the fly without closing and recreating the order.
- **Composable:** multi-order strategies automatically shift liquidity between linked orders as they are filled, reducing the cost of manually creating orders.
- **Re-usable:** tokens acquired in one order become available for trading in a linked order once markets move into range.
- **MEV-Resistant:** trading is resistant to sandwich attacks, the most common form of Maximal Extractable Value (“MEV”).

Background

DEXs are protocols that allow for tokens to be exchanged between users in a permissionless manner. The prototypical example is a smart contract (“liquidity pool”) that collects crowd-sourced liquidity from users (“liquidity providers”) and issues a receipt token (“pool token”) in return. The composition of the liquidity

pool may change over time as traders exchange their own tokens for those inside the pool.

An important mainstay of DeFi is the invariant-function DEX, where exchange rates are determined by equations that force the composition of the liquidity pool to adhere to a predefined profile (“bonding curve”). Importantly, liquidity providers on existing DEXs are beholden to the parameters of the liquidity pool to which they contribute their tokens. Thus, the agency of an individual liquidity provider to decide their own exchange rates and execute precise trading strategies is suppressed in favor of a prescribed, generic interpretation of their intentions.

The restrictive nature of existing liquidity pools largely stems from the fact that each pool and its composite liquidity positions are governed by the same bonding curve in either direction. In other words, the same curve is used for both buying and selling, and, ignoring fees, tokens sold by a liquidity pool may be repurchased by the pool at the same exchange rate. However, many common trading strategies – including limit orders and range trading – are “asymmetric” in nature, involving independent buy and sell patterns that are irreversible after being executed. Without the necessary infrastructure, these strategies remain inefficient and largely unavailable in DEXs.

Various DEXs have devised workarounds, for example Uniswap v3 allows users to mimic limit orders by providing liquidity in a thin out-of-range price interval that is traded when prices go through the associated interval. However, in practice, such limit orders are reversed when markets retrace. To abate limit order reversal on Uniswap v3, users motivated by this use-case must constantly monitor the state of the system and remove their position immediately upon execution. These tasks require specialist tooling and incur additional overhead costs, including and

especially gas considerations. Similar constraints emerge in trying to perform ‘buy low, sell high’ range trading strategies in existing concentrated liquidity pools, as offering liquidity in different ranges either requires separate liquidity positions or traders must automate costly transactions to close and recreate a liquidity position in their desired ranges as prices move.

The absence of sophisticated trading capabilities in DEXs partly explains the volume and liquidity discrepancy between DeFi and CeFi. Most trading volume still occurs on CEXs and a large percentage of that volume comes from automated strategies¹ which are inefficient or prohibitively expensive on DEXs. Since 2021, DEX growth has stalled in comparison to CEXs, with total DEX volumes falling to 11% of CEX volumes in September 2022, down from 18% in July². Declining DEX activity is exacerbated by the rise of MEV attacks in decentralized venues, which have further restricted liquidity from flowing into DeFi - and further emphasize the need for DEX infrastructure to evolve.

Asymmetric Liquidity

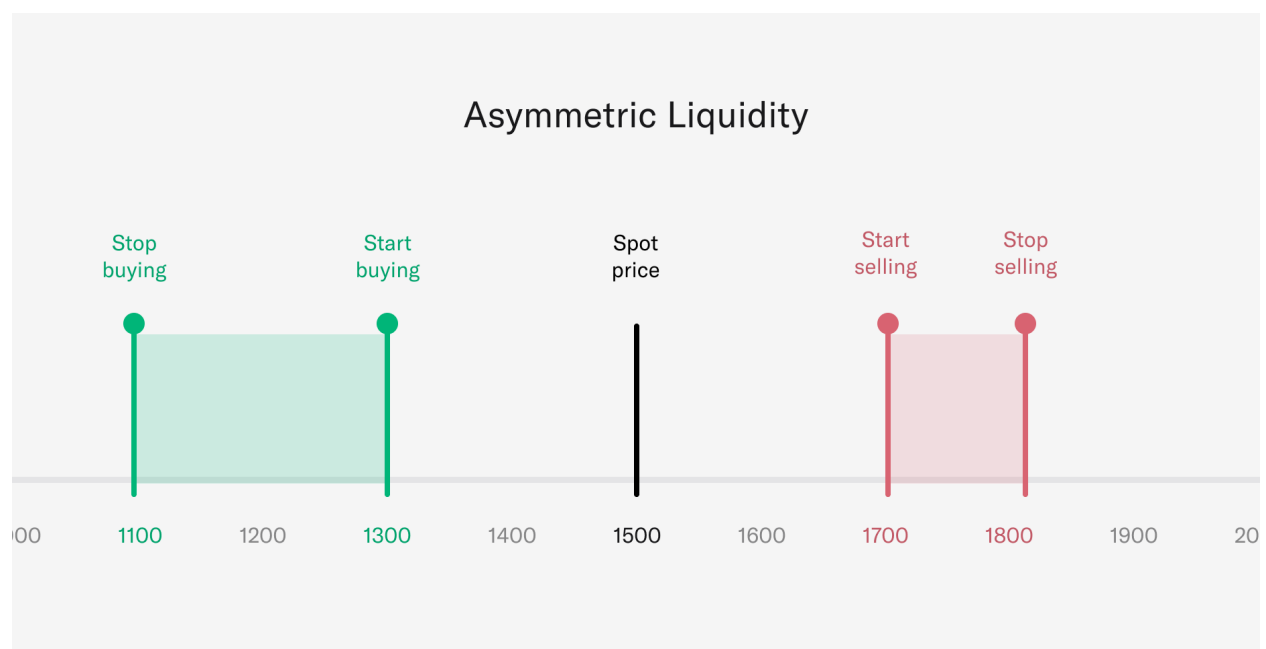
The defining idea of Asymmetric Liquidity is the rejection that on-chain market makers are necessarily neutral with respect to their market outlook. In a conventional liquidity pool, market makers are most successful when the relative valuation of the two assets comprising a bonded pair varies to only a small degree, and effectively trades sideways over long periods of time. The gambit is that a sideways trend remains relatively constant; divergence from constant valuation is the causative element of Impermanent Loss³.

¹ [*Market vs. Limit Orders*](#) (Harris and Hasbrouk, 1996): limit orders account for approximately 45% of total NYSE orders.

² [*Coinshares DeFi Spotlight \(Q3, 2022\) report*](#) shows declining DEX volumes vs. centralized exchanges.

³ [*Impermanent Loss in Uniswap v3 AMMs*](#)

Asymmetric Liquidity offers a radically new paradigm - one of on-chain liquidity where users execute personalized trading and market-making strategies. A defining characteristic of these strategies is their asymmetry: Whereas a conventional liquidity position uses a single bonding curve for both buying and selling, Asymmetric Liquidity allows for the creation of individual user strategies composed of two bonding curves, where each curve executes irreversible trades. In practice, this gives users the ability to create automated trading strategies composed of one or two on-chain limit or range orders for any given token pair, with each order represented by a unique, adjustable bonding curve.



A user strategy (composed of two range orders) buys ETH between 1100 and 1300 USDC and sells ETH between 1700 and 1800 USDC. The strategy continues buying or selling whenever prices cross into the predefined price ranges until liquidity in the orders is fully traded, or the strategy is updated/canceled.

The following sections describe key components and core functionality of the system:

- Adjustable Bonding Curves
- User Strategies
- Asymmetric Liquidity Pools
- MEV Resistance

Adjustable Bonding Curves

At the core of Carbon is a novel mathematical formula that can be used to create an array of bonding curve shapes, including constant sum, constant product and anything in between. A user controls their order conditions via a set of parameters (in addition to the token balance) which determine the curve's shape and exchange rate profile. See the formula for Carbon's adjustable bonding curve in [desmos](#).

In contrast to existing liquidity pools, as the curves underlying Carbon pools execute orders, acquired tokens cannot be repurchased by the pool if prices retrace. As a result, orders are, by design, irreversible. In the case of a single-curve strategy (e.g., a limit order), as the order is filled, acquired tokens are effectively taken “off-curve” and accessible only to the user who submitted the trade. In the case of a dual-curve strategy, tokens acquired in the first order become available for selling in the second order.

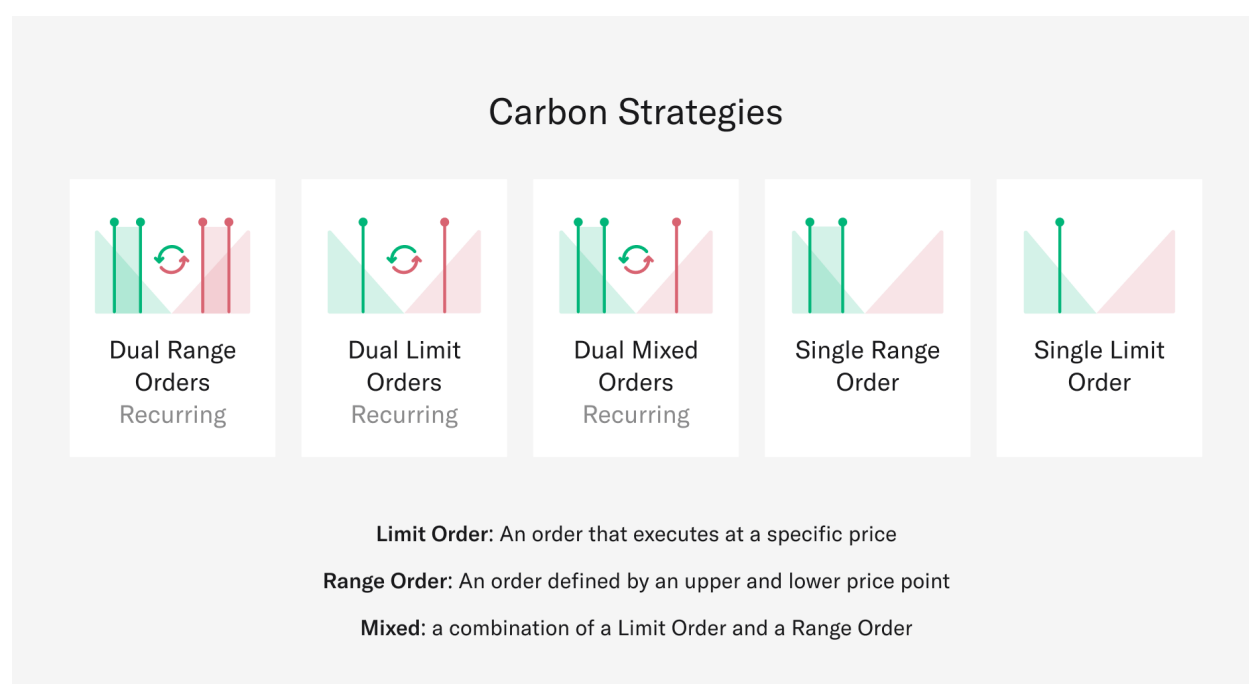
Carbon curves are also adjustable on-chain, allowing users to inexpensively change the conditions of a submitted order prior to execution. In existing concentrated liquidity pools, changes to a user's position (e.g., moving liquidity between price ranges) require closing and recreating the user's entire liquidity position. This is complex to manage, and more importantly, prohibitively expensive when trying to

react in real-time to market movements. In Carbon, adjustments to an order can be made “on the fly” – i.e., without a user needing to close and recreate their underlying liquidity position. Parameters are adjustable via low-cost transactions involving minimal computational overhead, making strategy management significantly more gas-efficient.

See the math & formulas underlying Carbon in the [patent filing](#).

Strategies

In Carbon, users create “Strategies” composed of one or two orders. Strategies and their composite orders remain open until they are canceled or filled. Dual-order strategies may be created with “linked orders” – meaning, tokens acquired from selling on one order become available for selling on the linked order. Linked-order strategies are recurring in that liquidity shifts between the two curves as prices move into either range. Theoretically, a linked-order strategy can trade indefinitely, banking profits on each trip back and forth between the two ranges.



Carbon Strategies are permissionless such that any user can create a Strategy for every standard ERC-20 token pair. Similar to liquidity positions in existing concentrated liquidity pools, Strategies are non-fungible in nature: When a user creates a Strategy, they receive in return a Non-Fungible Token (“NFT”) representing the Strategy.

Asymmetric Liquidity Pools

Each liquidity pool in Carbon is composed of all Strategies involving the associated token pair. For example, all Strategies involving both USDC and ETH make up the liquidity for the USDC/ETH liquidity pool. A matching and routing engine handles requests by individuals to trade “spot” against Carbon’s liquidity pools. Order Matching specifically refers to a direct exchange path through a single liquidity pool’s orders, while Order Routing refers to indirect exchange via an intermediate token, enabling exchange between two tokens where no live strategy exists.

MEV Resistance

An important issue for on-chain liquidity is Maximal Extractable Value or “MEV” - which results in profits extracted by parties who control the transaction flow, typically miners. A primary MEV attack vector for on-chain liquidity pools is the so-called “sandwich attack”, where a genuine transaction is sandwiched between transactions of the attacker.

An AMM sandwich attack is very similar to front-running in traditional markets, except that a sandwich attack is guaranteed to either succeed, or to fail costlessly.

The way it works is as follows:

1. The attacker identifies a reasonably large trade order, e.g., buying ETH against USDC; this order is a “market order”, i.e., it fixes a USDC amount, and takes whatever amount of ETH it will get.
2. The attacker inserts a similarly-sized order buying ETH against USDC immediately before the attacked transaction.
3. The attacker inserts an equal and opposite transaction to (2) immediately after the attacked transaction.

What happens if the above is executed successfully is that the price at which the attacked transaction gets filled is artificially high, due to the slippage introduced by the transaction (2). The transaction (3), which is now selling ETH, benefits from both the slippage introduced by the transaction (2) and that by the attacked transaction. In essence, the attacker shifts the transaction to a higher price point off-market and can pocket that difference in price in a risk-free manner.

This particular attack vector is closed in Carbon. While an attacker can still front-run a transaction as described under (2) above, the reverse transaction (3) is prohibited by the asymmetry of the system. Therefore, the attacker is no longer in a position to retreat from the consequences of their frontrunning trade, which defeats the underlying tenability of the sandwich attack entirely.

Carbon Implementation

The specifics of Carbon require Bancor DAO governance approval and are subject to change, with voting occurring via Carbon’s proposed governance token, vBNT. As implementation of Carbon evolves via community discussions, this section will be adjusted and expanded accordingly.

Protocol Fees

Carbon has flexible fees that can be collected when operations are performed by its users. Governance controls the activation and destination of fees collected by the Carbon protocol. Ultimately, it is up to governance to determine which fee switches are enabled, the fee amount associated with each operation and the destination of collected fees. Initially, proposed fees can include:

Taker Fee:

Executed Spot Trades are charged a DAO-set percentage fee taken from the destination token.

Maker Fees:

Executed Strategies are charged a DAO-set percentage fee taken from the destination token. A fixed strategy creation fee may be enabled by the DAO and collected when strategies are created or updated.

Use Cases & Simulations

Here we present several on-chain strategies that are planned to be supported in the first iteration of Carbon. Note that these strategies are supported natively on Carbon – i.e., execution requires no third-party oracles or keepers.

The use cases below are simulated in the [Carbon Simulator](#).

Limit Order

Bob creates a strategy where he buys ETH when the price goes to 1500 USDC. He is willing to spend 5000 USDC for this. He would like to buy at this specific price only.

Buy Low, Sell High Strategy

Alice creates a strategy where she buys ETH as the price goes between 1500 and 1600 USDC. She is willing to spend 5000 USDC for this. She also wants to sell her acquired ETH when the price goes from 2000 to 2500 USDC.

Average-In Order

Jane creates a strategy to buy ETH between 1600 and 1500 USDC when the current price is at 2000 USDC. She is willing to spend 5000 USDC for this.

Token Distribution

Token project XYZ creates a strategy to sell 1 million units of its XYZ token in the price range of 0.50-2.00. The project would like to “back-load” the distribution process, such that half of the tokens are sold between 1.50-2.00. As a result, the project will receive roughly 1.37m in cash for its tokens once they’re all distributed.

Conclusion

Carbon redefines the capabilities of on-chain liquidity pools by introducing Asymmetric Liquidity with Adjustable Bonding Curves as core features. The protocol supports the permissionless creation of on-chain orders that can be linked together, adjusted on-the-fly and are irreversible on execution, while enabling trading that is resistant to MEV sandwich attacks. There is no Impermanent Loss in Carbon, in the sense that orders are not buy-and-hold liquidity positions, but the expression of a particular trading view.

At a time when most DeFi yields have dropped below U.S. Treasury bond yields, and centralized finance still retains the vast majority of crypto liquidity and trading

volume, Carbon equips users with new ways to trade in DeFi while supporting on-chain, decentralized liquidity for the token economy.

Users can achieve greater efficiency, flexibility and precision to execute personalized DeFi strategies, including native limit orders, linked range orders, ‘average in’ orders and token distribution strategies. However, this is only the beginning: Asymmetric Liquidity with Adjustable Bonding Curves expand the design space for on-chain liquidity and open the doors to a wide range of future DeFi applications and products.

APPENDIX

Glossary

Item	Description
Order Maker Order	<p>An Order (to be understood as a “Maker” Order) corresponds to a single curve for trading in one specific direction on a single pair of tokens. An Order remains live until canceled.</p> <p>An Order can be at one specific price (buy at x price), or in a range (buy between x and y).</p>
Linked Orders	<p>Linked Orders are orders where the tokens purchased in one Order become available in the other one. For example, one of the orders may lead the AMM to sell USDC for ETH and the other one ETH for USDC. Whenever USDC is sold by Order #1, the acquired ETH is made available for sale on Order #2.</p>
Strategy	<p>A Strategy consists of a single Order or multiple “linked” Orders on a single token pair, created by a single user.</p> <p>Each Order corresponds to its distinct underlying bonding curve. It is possible to have zero, one or multiple Orders, and therefore curves, within a Strategy. When two Orders are part of a Strategy they are Linked Orders.</p>
Asymmetric Liquidity Pool	<p>The combination of all Strategies of all users on a single token pair.</p> <p>For example, all Strategies involving both ETH and USDC make up the liquidity for the ETH/USDC liquidity pool. Individuals trade “spot” against the network’s liquidity pools.</p>
DEX / Liquidity Network	<p>The combination of all pools across all pairs.</p>
Trade Taker Order	<p>A Trade is a Taker Order that trades against the current DEX, via the matching algorithm. A Trade is executed immediately, or fails (“fill-or-kill”).</p>
Adjustable Bonding Curve	<p>A bonding curve that can be changed on-the-fly without needing to close and recreate its associated liquidity position.</p>
Matching	<p>Matching a Taker Order / Trade request to one or more Maker Orders within a single liquidity pool.</p>
Routing	<p>Optimally routing a Taker Order / Trade request within the entire liquidity network, possibly involving an intermediate token and two or more “hops” between pools.</p>
Price Impact	<p>The change in token price directly caused by a Trade. Price impact is reflected as the difference between the current market price and how the Trade impacts the total liquidity in a pool.</p>
Slippage Price Slippage	<p>Slippage is reflected as the difference between the current price for a marginal transaction on the pool, and the price at which the Trade is actually executed.</p>

Resources

1. [Carbon Website](#)
2. [Carbon Litepaper](#)
3. [Carbon Whitepaper](#)
4. [Carbon Patent Filing](#)
5. [Carbon Simulator](#)

CARBON

carbontdefi.xyz