Term Structure AMM

Bringing yield curves on-chain

Alberto Molina alberto@io.builders
Miguel Gómez miguel.carpena@io.builders
Roberto Delgado roberto@io.builders

June 24, 2025

Abstract

The majority of the bonds market traded volume occurs in the OTC market, where the liquidity is fragmented among a handful of dealers. This fragmentation, combined with opaque RFQ mechanics and limited pre- and post-trade transparency, leads to wide bid/ask spreads, high inventory-carrying costs, settlement delays, and significant regulatory and custody frictions. Some bonds might not trade for weeks, forcing institutional players to rely on proprietary valuation models to estimate fair value. Term Structure AMM tackles these inefficiencies by providing a constant liquidity source—aggregating fragmented demand into on-chain pools, automating execution to reduce spreads and inventory/settlement risk, and embedding compliance to stay regulatory compliant. By removing manual negotiations and intermediaries, we deliver instant, cost-efficient transactions and fair, competitive pricing.

Introduction

Currently, AMMs primarily facilitate trading between different currencies, either through token pairs or weighted pools (e.g., Balancer). These pools employ various pricing models depending on the asset type, with the most widely used being the constant product formula, popularized by Uniswap.

$$x \cdot y = L^2 \tag{1}$$

Other models, such as StableSwap's invariant, are optimized for assets with low price volatility—like stablecoin pairs or pegged tokens (e.g., stETH/ETH). All these models lack of inclusion of intrinsic asset value and rather price the risky asset in terms of the numeraire based on order flow.

Our approach represents a big shift compared to how price discovery works in vanilla AMMs. AMMs for token pairs predominantly rely on a complex sequence of auctions [1] where solvers, market makers, searches and block builders compete to offer the best trade execution price for users. During the oder flow auction, searchers seek to profit from price signals aligning the price of the pools with that of external deeper liquidity venues.

Bonds are less liquid assets due to their heterogeneity and carry some associated market risk, therefore the same arbitrage strategies cannot be applied in this market. This difference in the pricing mechanics of this types of asset with respect to FX or cryptocurrencies makes them less suitable for vanilla AMMs. Our algorithm instead of purely relying on trades for price discovery, it incorporates a fair value component that is updated every block based on the idiosyncracies of the bond, the characteristics of the issuer and other market data such as the interest rates regime.

Our solution aims to bridge a TradFi concept to the DeFi space bringing a new wave of institutional investors. In order to achieve this we need to make sure we stay regulatory compliant and respect some important TradFi rules. We achieve this leveraging our Hedera open-source project Asset Tokenization Studio (ATS). [1] We use this toolkit for the issuance of bonds under the ERC-1400 standard.

Previous Work

The concept of a bond AMM was already explored in a prior Uniswap V4 Hookathon cohort, though the approach differed significantly from ours. In their project, the hook handled both minting and burning bond positions, merging primary and secondary markets.[2] In contrast, our hook establishes a clear separation between primary and secondary market and it aims to be a more institutional solution.

TermMax has also researched similar AMMs, specifically for their FT and XT tokens, where the price adjusts as the market approaches expiration. Their model uses concentrated liquidity but remains under development.[3]

Bunni is another relevant project, enabling custom liquidity curves through Uniswap V4's no-op hooks—a feature our solution also leverages.[4]

[1] `https://hedera.com/asset-tokenization-studio`
[2] `https://github.com/0xLighthouse/signals/pull/31`
[3] `https://docs.ts.finance/`
[4] `https://docs.bunni.xyz/docs/v2/overview/`

Algorithm

Our algorithm aims to replicate conventional market makers bond-pricing mechanics. Unlike FX instruments, bonds—while sharing some characteristics—require different considerations. Bonds are valued using a mix of idiosyncratic factors and market dynamics; the Discounted Cash Flow (DCF) model is a common baseline.

In our framework, for this first iteration, we focus on a zero-coupon bond whose liquidity is distributed across the entire pool range. We split pricing into two components: a "fair valuation" leg via traditional DCF, and a "market dynamics" leg that updates with each new block. The fair value is calculated based on the face value F, the time to maturity θ and the yield of the bond r.

$$\hat{p}_t = \frac{F}{(1+r_t)^{\theta_t}} \tag{2}$$

The yield to maturity is obtained from an EigenLayer AVS where a set of operators run an off-chain script that retrieves information from trusted sources. It will be estimated based on a combination of factors such as the interest rate regime, the credit rating and the bond's covenants.

On blockchains like Ethereum (~ 12 s block time), new price data arrives each block. Let $p_{t,\text{bottom}}$ be the on-chain closing price of the last block. At the block at time $t+1$, we blend the DCF price with the previous block's closing price, using a sensitivity $\delta_{p,t+1}$ that determines how much of the market information component we want to include in the calculation.

$$p_{t+1,\text{top}} = \hat{p}_{t+1} + \delta_{p,t+1}(p_{t,\text{bottom}} - \hat{p}_{t+1}) \tag{3}$$

Setting $\delta_{p,t+1}$ to 1 in Eq. 3 will mean the pool is operating as a vanilla AMM, while for a value of 0 is purely following the DCF.

δ_p is updated each block based on trading volume above a noise threshold, using a second order price sensitivity γ_p.

$$\delta_{p,t+1} = \delta_{p,t} + \gamma_p |\Delta p_{t,\text{market}}| \tag{4}$$

Using a noise threshold allows the pool to get rid of meaningless trades driven based on individual decision rather than market information. Once updated the sensitivity, it is then linearly reduced at a constant rate per block until reaching normal operation levels.

By measuring price changes from the top to bottom of each block (instead of per transaction), we prevent arbitrageurs from splitting large trades into small ones to manipulate δ_p.

The price needs to also be aligned with the reserves volume as per the standard Uniswap V2-style formula (Eq. 1). We assume here x are the bonds and y the denominated currency.

$$\frac{y}{x} = p_{t,\text{top}} \tag{5}$$

Under fixed liquidity, Eq. 1 and Eq. 5 can't both hold if $p_{t,\text{bottom}} \neq \hat{p}_{t+1}$, so we relax "constant-reserves" between blocks, resulting in inter-block surpluses.

$$\Delta_x = x_{t,\text{bottom}} - \frac{y_{t,\text{bottom}}}{p_{t+1,\text{top}}} \tag{6}$$

$$\Delta_y = y_{t,\text{bottom}} - x_{t,\text{bottom}} \cdot p_{t+1,\text{top}} \tag{7}$$

$$\text{surplus} = \max(\Delta_x, \Delta_y) \tag{8}$$

We temporarily hold that surplus "off-book" and recycle it when new LPs join. While idle, it is deployed into yield-farming strategies to generate extra income for LPs.

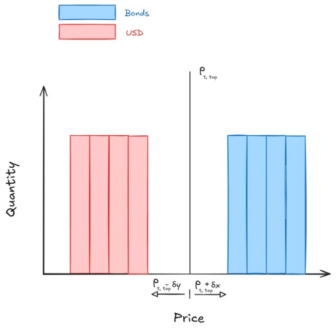
LPs can provide single-sided liquidity. For example, when there's surplus bond inventory, an LP may deposit only token y. The hook automatically pairs it with token x per the constant product rule (Eq. 1). Since these LPs supply only half the required assets, they receive half the LP tokens.

The diagrams in Figure 1 illustrate the AMM state transitions across blocks. The figure shows the AMM at the start of a block with evenly distributed liquidity. As bond purchases increase, the price rises, transitioning to the state in Figure 1b.

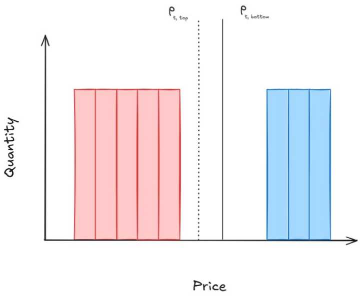
While intra-block trading follows standard AMM mechanics, this does not accurately reflect bond pricing. In theory, bond value shouldn't shift from trading volume alone, as its price is derived from a fixed repayment commitment. To address this, we reset the pool price at the top of each block to align with our model as defined in Eq. 2.

In the example above, this requires lowering the bond price, which means temporarily removing some USD from the pool. Figure 2 shows how the pool manages this excess reserve: surplus USD is removed, held separately, and rehypothecated to generate yield for LPs. As single-sided LPs deposit bonds, the idle USD is gradually reintroduced.

This structure models the price impact of trades within a block, capturing demand-driven market dynamics. At the next block, it incorporates the fair valuation component "kicking" the price by removing surplus liquidity—mirroring how additional sell orders appear in CLOBs in resilient markets to profit from the bid-ask spreads. However, in our case, this adjustment happens only at discrete block intervals, and we align the price by removing liquidity from the opposite side.

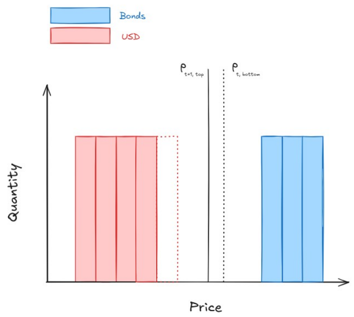


(a) AMM at start of block (even liquidity).



(b) AMM after bond purchases (price↑).

Figure 1: Price movement from the top to the bottom of the block at time t



(a) Liquidity distribution after rebalance

(b) Liquidity distribution after single sided deposit

Figure 2: Liquidity provision mechanics at time t+1

Non-PBS Compatible Networks

The pricing discovery mechanics in vanilla AMMs previously described, rely on mechanisms such as Proposer Builder Separation (PBS) and Maximum Extractable Value (MEV) to align the value of the pool with that of deeper liquidity venues. Not every network natively supports it, using this algorithm that prices assets not only based on trades would allow deploying pools on networks such as Hedera where its gossip-about-gossip protocol does not support intra-block transaction ordering. This will further enable new participants to join the ecosystem.

Market Integration

The prototype developed goes beyond being a custom AMM, it supports the full lifecycle of the bonds. By integrating the hook with the ATS, issuers can deploy their bonds, manage the issuance through its interface and then come back to the hook after the subscription period has concluded. The moment the issuer launches the pool the secondary market is open for trading.

Regulatory compliance is another important consideration, the ATS conforms with multiple requirements, such as the presence of a control list and metadata associated with the applicable regulation of the bond.

To bring major institutional players, the algorithm would need to be integrated within a regulated market infrastructure. There are some examples such as OpenBrick/OpenX for the spanish market that+ provide a comprehensive suite of services for the registration of investors and supports legal figures such as the ERIR (Responsible Entity for Registration and Recording), fulfilling requirements under Spain's law.

Partners Integration

Out of all the partners present in this cohort we decided to integrate EigenLayer and Circle since we considered their services added the most value to our idea.

EigenLayer

The AVS is a core component in the ecosystem, we needed a way to bring economic data from off-chain sources backed by crypto-economics. Operators will run a script that performs complex calculations fetching data about the bond from external sources.

Circle

We use the compliance engine for the screening process. As soon as users connect to the application their status is checked and if they are not compliant they are added to the ATS blacklist restricting further interactions with the pool.

Another long term solution we would like to explore as integration that is out of the scope of the MVP is using Circle Mint for the mass payments upon bond expiration.

Limitations

This novel mechanism is not without its shortcomings that should be considered.

Arbitrage

Accurate parameter calibration is essential to mitigate the risk of exploitation by arbitrageurs seeking to profit from pricing inefficiencies. Although fixed-income arbitrage typically presents fewer opportunities than FX markets, it is critical to safeguard the pool against edge cases that could erode LP returns.

Compared to FX, bonds are inherently less liquid due to their heterogeneity. Fixed-income arbitrage is often described as "picking up nickels in front of a steamroller," highlighting the strategy's limited margin and high risk. However, empirical research such as the work in [2] demonstrates that portfolios constructed using these strategies can achieve substantial positive returns. These findings also note a growing adoption of fixed-income arbitrage among hedge funds.

Our pool introduces the market sensitivity parameter, δ_p, which plays a dual role: it incorporates real-time market dynamics into the pricing model and serves as a defensive mechanism against arbitrage exploitation. Calibrating δ_p appropriately is crucial—when arbitrage activity intensifies, a larger share of the pool's pricing should be driven by prevailing market dynamics. The optimal value of δ_p will depend on a combination of bond-specific idiosyncrasies and broader macroeconomic conditions.

Atomic arbitrage opportunities may emerge as well when external liquidity sources offer the same bonds at different prices. Future iterations of the model will focus on empirical parameter optimization using historical market data and rigorous backtesting against out-of-sample datasets to validate model performance under various market conditions.

Reduced Liquidity During Rebalancing

Fair valuation is achieved at the cost of temporarily shallower liquidity, resulting in increased price slippage for traders. Additionally LPs expect their whole tokens to be generating income so we cannot simply freeze them. To deal with these two issues we rehypothecate the idle funds using yield farming strategies and offer single-sided LP incentives for those who come first and redeposit the idle funds into the pool. Additionally, when LPs remove liquidity, we first look if the vaults have enough liquidity to repay them, minimizing this way the time the funds are idle.

Unfair Execution for Low-Fee Transactions

Traders using low gas fees may execute trades further from fair value, especially in volatile conditions. This can result in "gas races", disproportionately benefiting validators—contrary to our goal of maximizing value for our stakeholders (LPs and swappers). To mitigate this we use a dynamic fee that lowers from top to the bottom of the block.

References

[1] Angela Lu. "Illuminating Ethereum's Order Flow Landscape". In: *Flashbots Writings* (Jan. 2024). URL: `https://writings.flashbots.net/illuminate-the-order-flow` (visited on 06/22/2025).

[2] Jefferson Duarte, Francis A. Longstaff, and Fan Yu. "Risk and Return in Fixed Income Arbitrage: Nickels in Front of a Steamroller?" In: *Review of Financial Studies* 20.3 (May 2007), pp. 769–811. DOI: `10.1093/rfs/hhl026`.