Financially-Stable Automated Market Making for Decentralized Fixed-Rate Lending and Trading

***Abstract*—Compared to typical crypto-exchange AMMs, fixed-rate AMMs are much more difficult to design due to the inter-dependency of interest rate, loan price, and, especially, time factor. The literature lacks scientific developments. Existing efforts are either too simplistic, focusing more on automation and less on economic effect, or too engineering-driven, using ad hoc formulations without economic justifications. The fact is that today's fixed-rate DeFi is struggling to attract any side of participation, whether it is the borrowers, the lenders, and the liquidity providers (LPs). We propose BondMM, a new fixed-rate AMM algorithm as an effort to advance the state of the art. It has the mechanics of a constant-function AMM with path-independence property, which is desirable for avoiding money-pumping attacks. It is constructed, whose correctness provable, using elegant closed-form mathematics, hence efficient for blockchain implementation. The properties and performance of BondMM are validated with our theoretical analysis and simulation-based evaluation. Specifically, we show that today's benchmark fixed-rate AMMs are either too financially-risky (Yield AMM) or too trader-expensive (Notional AMM) to sustain practically for the long term. BondMM is superior to both regarding properties that define the financial strength of a lending institution: 1) better price impact to attract borrowers and lenders, 2) better capital efficiency to attract LPs, and 3) better financial stability for both participants and LPs.**

***Index Terms*—Decentralized Finance, Blockchain, Decentralized Computing, Automated Market Makers.**

I. Introduction

Fixed-income markets are the lifeblood of the global economy. They are more than twice the size of the global stock market. As blockchain technology [1] is disrupting the financial sector, we expect to see more and more decentralized finance (DeFi) products in the near future, and if the pattern in traditional finance applies, fixed-income DeFi should be one of the leading players in the DeFi space, if not the leader. How far are we getting there? Today, the DeFi lending sector is about \$45B, which is 3% of the overall \$1.5T crypto market capitalization. In particular, few fixed-income products exist [2]–[10], with a total value locked (TVL) less than \$1B, a minuscule 0.07% of the crypto market.

In other words, the market remains huge and wide open for fixed-income DeFi. We approach this opportunity from two different angles: 1) is it because existing lending protocols are not well designed for fixed-rate offerings, thus struggling to attract adoption? and 2) is it because there are just too few products available, suggesting room for more to enter the market? The answer is yes to both. In this paper, we propose a new fixed-rate lending and trading protocol based on automated market making (AMM) [11]–[14] that brings advances to the state of the art.

Related Work and Limitations. DeFi lending leaders such as AAVE [4], Compound [5], and MakerDAO [15] are designed for variable rates. They do offer fixed rates but only as an add-on product for borrowers, and the rates are so high that they account for only 1% of the total borrowing volume. This is because the lending pool needs a guaranteed spread to profit lenders at the cost of fixed-rate borrowers.

On the other hand, few protocols exist exclusively for fixed-rate lending and, unfortunately, they remain unattractive to the liquidity providers (LPs). The major protocols are Yield Protocol [6] and Notional Finance [7]. Based on such a protocol, one can develop fixed-income applications in different forms, for example, principal-and-yield split products with Pendle [8], Element [10], or Swivel [3], and structured products with BarnBridge [2]. In theory, these protocols operate as an AMM using the same rate for borrowing as that for lending. As the LPs lose income from bid-ask spreads, they must make money only from transaction fees. Income from fees is significant only if there is a lot of secondary trading, not just primary lending and borrowing transactions. Unfortunately, today's products offer very short maturity duration (3-month, 6-month, and 1-year), which is not that inviting to secondary traders due to lack of leverage. Trading with speculation is the main reason as to why the bond market is many times larger than the traditional lending market. Speculating traders like long-duration bonds because of the increased leverage for their bets. For example, a bond of 10-year maturity offers a leverage more than 10 times that of the 1-year due to compound effect.

Another drawback of existing fixed-rate AMMs is limited capital efficiency, most noticeably with Yield [6]. Their mathematical formulas to ensure continuous liquidity for every possible trade are not well designed, leading to liquidity excessively reserved for cases of extreme prices and interest rates, which, however, rarely happen. Better, we should concentrate liquidity distribution on realistic price/rate ranges. Notional [7] is the first to try to achieve this, but their pricing is crafted in ad hoc engineering manner to profit the LPs greedily, which incurs bad rates for the service users (borrowers/lenders/traders). This is an extreme approach because, economically speaking, no user would come to Notional due to high costs.

Last but not least, all existing works on fixed-rate AMM are product-based, engineering efforts, and lack verifiable scientific proofs. It is no surprise that their fix-rate maturity is kept short because this is a safe approach for the LPs, so that borrowing and lending positions can be settled quickly to avoid long-term risks. As aforementioned, this is not desirable to bond traders because they prefer long maturity.

1

Our contributions. Fixed-rate AMMs need deep liquidity (we need LPs) and large trading volume (we need borrowers/lenders/traders). We should not make the LPs happy simply by charging the users more. It is ideal to have a Nash equilibrium pleasing all sides maximally. Our ambition is not to find such an equilibrium; this can be a non-tractable problem computationally. Instead, we aim to do better than today's solutions in the most important aspects.

Specifically, we propose BondMM, a new fixed-rate AMM algorithm that offers the following desirable properties: 1) better price impact to attract borrowers and lenders, 2) better capital efficiency to attract LPs, and 3) better equity stability toward short-term liquidity coverage and long-term solvency. Regarding the third property, a high equity means better ability to fulfill immediate transactions and resolve obligated debts in the long term. However, too high an equity means too much profit for the LPs at the cost of the users, which we should avoid. Because the business model of zero-spread fixed-rate AMMs is driven by transaction fees, we should keep LP equity as stable and close to the initial contributed capital as possible.

In terms of construction, BondMM is a constant-function AMM with path-independence property. This is desirable for avoiding money-pumping attacks where one strategizes a series of transactions to pump and dump to exhaust pool liquidity. It is formulated, whose correctness provable, using elegant closed-form mathematics, hence efficient for blockchain implementation. The properties and performance of BondMM are validated with our theoretical and experimental analyses using both real-world and synthetic datasets. The results show that BondMM substantially outperforms Notional and Yield. Our work enhances the currently-thin literature on fixed-rate AMMs and can serve as a verifiable scientific benchmark.

The remainder of the paper is organized as follows. Essential background on fixed-rate AMM and other preliminaries for our work are presented in Section II. BondMM is introduced and described in detail in Section III. The evaluation results are analyzed in Section IV. The paper concludes in Section V with pointers to our future work.

II. PRELIMINARIES

We are interested in loans of fixed rate having a finite maturity. Someone borrowing $b = \$100$ at interest rate $r = 5\%$ yearly for $T = 10$ years has to pay back a total of $a = b \cdot (1+r)^T = \$162$, which consists of the principal amount b and interest amount $a - b = \$62$. Depending on products, these amounts can be collected by the lender in smaller periodic payments or as a lump sum at maturity time. The latter applies to zero-coupon bonds where the interest total is prepaid in advance at the beginning and the principal total is one single future payment. Each bond has a "face value" equal to the principal amount, which is materialized only at bond expiration. Until then, bonds can be traded at floating price driven by supply and demand. The interest prepay has the same effect has pricing the bond at a discount from the face value. Whoever holds a bond at maturity will be paid its face-value amount. In what follows, for the sake of theoretical formulation, we assume the zero-coupon bond model to represent fix-rate loans. For implementation in practice, this model can easily be adapted for other repayment methods, for example, distributing yields over the time.

A. *Fixed-Rate AMM*

Consider bonds that mature at some time T in the future. Bonds are implemented in the form of a fungible token. Let us call it "bond token", or simply "bond". Money value is in the form of some numeraire "cash", say DAI. Without loss of generality, let the bond face value be 1 DAI.

At any time instant, a fixed-rate AMM is associated with the tuple (x, y, r, p) where x, y are the respective amounts of bond and cash in the liquidity pool, and r is the reference interest rate (annualized). The reference rate is what the participants sees instantaneously as representative of the AMM market. In practice, a spread between the lending rate and borrowing rate can be introduced, but for the the sake of theoretical presentation, assume no rate spread and so r is the marginal interest rate for both lending and borrowing. This corresponds to a marginal bond price p. In theory, bond rate and price have the following relationship, $p = \mathrm{e}^{-tr} \Leftrightarrow r = \frac{1}{t}\ln\frac{1}{p}$, where t is time to maturity. The actual bond price in a trade should depend on the order size, i.e., the number of bonds traded in the order. Typically, r and p are computed from the pool reserves (x, y). Different AMM designs have different formulas for them.

Transactions. There are four types of actions from the user: bond minting (borrowing), bond burning (borrower exit), bond buying (lending), and bond selling (lender exit). They all result in either adding bonds (to get cash out) or adding cash (to get bond out) to the pool. Therefore, without loss of generality, we represent each transaction by a pair $(\Delta x, \Delta y)$ where $\Delta x > 0$ (or $\Delta y < 0$) means bond selling and $\Delta x < 0$ (or $\Delta y > 0$) means bond buying. This transaction will result in a a change

$$(x, y, r, p) \to (x_{new} = x + \Delta x, y_{new} = y + \Delta y, r_{new}, p_{new}).$$

Given an order (Δx, Δy), which is input as an amount in ether Δx or Δy, the formula to calculate the other amount is according to the specific pricing model of the AMM.

B. *Existing Models*

To date, there are two main models: the Yield Protocol model [6] and the Notional Finance model [7].

1) Yield Protocol Model: Constant-function AMM [12]–[14] is the standard for decentralized cryptocurrency exchanges, e.g., Uniswap [16] and Balancer [17]. This approach has been applied to fixed-rate protocols, including Pendle V1 [8], Apwine (now Spectra) [18], and Yield Protocol [6]. The most noticeable one is Yield, which has a constant-power-sum invariant,

$$F(x, y) = x^{1-t/T} + y^{1-t/T} = C$$

where C is a constant over all transactions happening at time to maturity t. As a result, the reference price is

$$p = \frac{-dy}{dx} = \left(\frac{y}{x}\right)^{t/T} = \phi^{-t/T} \tag{1}$$

2

and interest rate

$$r = \frac{1}{t}\ln(1/p) = \frac{1}{T}\ln\frac{x}{y} = \frac{1}{T}\ln(\phi). \tag{2}$$

Here $\phi \triangleq \frac{x}{y}$ denotes the bond-to-cash ratio in the liquidity pool. Yield's critical drawback is low capital efficiency. Indeed, when there is less bond than cash in the pool, i.e., $\phi < 1$, the AMM formula results in a bond price exceeding the face value and, equivalently, a negative interest rate. No lender would then join. As such, it is a waste to reserve much LP capital to cover this unrealistic case.

2) Notional Finance Model: Similar to Uniswap V3 [19] concentrating liquidity on a realistic price interval, Notional Finance [7], followed by Pendle V2 [20], use a fixed-rate AMM that concentrates liquidity around an anchor rate. Its rate function is an extension of the constant-power-sum rate,

$$r = \kappa \cdot \ln(\phi) + r^*, \tag{3}$$

where r^* is the anchor rate and constant κ is a coefficient to tune the liquidity concentration. A lower κ means higher concentration, thus more flattening price curve. In the non-scaling case where $\kappa = 1/T$ and $r^* = 0$, Eq. (3) is precisely the rate formula of Yield Protocol.

Notional AMM expresses interest rate r as a simple interest: $p(1 + rt) = 1$. So, the reference price is

$$p = (1 + rt)^{-1} = \left(1 + t\kappa \cdot \ln(\phi) + tr^*\right)^{-1}. \tag{4}$$

However, instead of relying on the mathematics of a constant-function invariant, Notional has a crafted formula to price a trade order. To compute Δy given Δx, let

$$\overline{\phi} = \frac{x + \Delta x}{y - \Delta x}.$$

The average price for the order is set to

$$\overline{p} = \left(1 + t\kappa \cdot \ln(\overline{\phi}) + tr^*\right)^{-1}, \tag{5}$$

resulting in $\Delta y = -\overline{p}\Delta x$.

A side effect of this pricing design is that the trade price $\overline{p}$ is always more expensive then the AMM's new reference price p updated after the trade. Especially, this results in an expensive trade price for bonds of long maturity, thus unattractive to traders. Also, Notional's pricing curve is no longer being path-independent. Ideally, every AMM should be path-independent [14]. Another drawback of Notional's trade price formula is the difficulty of deriving Δx if the order is input in terms of Δy. There requires solving a non-trivial equation which has no closed-form solutions. This could explain why, as of now, Notional does not allow orders to input Δy. As later shown in our evaluation, by being greedy on the LP side, Notional would benefit the LPs in the long term but due to expensive price impact, it suffers from liquidity risks and solvency risks in the early phase of a bond's life.

III. BondMM: New Solution

In existing models, the AMM state is (x, y) and the interest rate is a time-independent function of the bond-to-cash ratio in the pool expressed in the count of bonds x,

$$r = R(\phi), \phi = \frac{x}{y}.$$

We propose a change. In our fixed-rate AMM, named BondMM, the state is (X, y) where X is the present value of bonds in the pool

$$X = xp = xe^{-rt}. \tag{6}$$

and the reference rate is a time-independent function of the bond-to-cash ratio expressed in the bond value X,

$$r = R(\psi), \psi = \frac{X}{y}. \tag{7}$$

The corresponding reference price at time-to-maturity t is $p = e^{-rt}$. Our intuition is that, from the perspective of economics, we should use a cash value (X) instead of a count value (x) to represent the financial state of the AMM. From the perspective of mathematics, we can avoid mathematical singularity cases that exist in Yield and Notional modeling. We also hypothesize that our rate formulation will result in better capital efficiency and equity stability for the LPs. This is confirmed in our evaluation to be discussed later.

State (X, y) is changed only if triggered upon a transaction. When BondMM is at state (X, y), the pool has $x = Xe^{rt}$ bonds at time-to-maturity t. The rate remains unchanged over the time when no transaction occurs, which is a desirable feature for avoiding arbitrage. During this period, while X does not change, x decreases automatically with time to always maintain the relationship $x = Xe^{rt}$. This is done by simply burning a corresponding number of bonds in the pool. The burning rate is the same as the current interest rate.

In short, BondAMM is summarized below:

$$\textbf{state } (X, y) \rightarrow \textbf{reference rate } r = R(X/y)$$

$$\xrightarrow{\text{at time-to-maturity } t} \begin{cases} \textbf{reference price} & p = e^{-rt} \\ \textbf{bond quantity} & x = Xe^{rt} \end{cases}$$

In what follows, we present the details for our AMM.

A. Rate and Price Design

We formulate the reference rate as the following logarithmic function of the bond-to-cash ratio:

$$r = R(\psi) = \kappa \cdot \ln(\psi) + r^* = \kappa \cdot \ln\left(\frac{X}{y}\right) + r^*. \tag{8}$$

Here, constant $r^* \in (0, 1)$ is the anchor rate, which should be chosen to reflect the long-term market mean, and constant $\kappa \in (0, 1)$ is the price-volatility factor to tune the concentration of liquidity on the price curve. Note that our rate function is similar to Notional Finance in the use of the log form, but the crucial difference is that we use the bond-value proportion $\psi = \frac{X}{y}$ instead of bond-quantity proportion $\phi = \frac{x}{y}$. Another major difference is in how we design price functions. We

3

will see below that the mathematics behind BondMM enables computation with elegant closed-form expressions.

BondMM is initialized as follows. Suppose that we want to set up the AMM with initial cash y_0 DAI and initial rate r_0. We set r^* to r_0 by default, but in practice can choose any anchor rate that represents an equilibrium state. In the default case, $r^* = r_0$, and so we need $X_0 = y_0$. Because $p_0 = e^{-Tr_0}$, a quantity of $x_0 = X_0/p_0 = X_0 e^{Tr_0}$ bonds will be minted. BondMM has the following properties relating the state, pool reserves, rate, and price.

Lemma 1. *Let t be time-to-maturity. The reference rate and reference price are related to the pool reserves (x, y) as follows:*

$$r = \frac{1}{1+\kappa t}\left(\kappa \ln \frac{x}{y} + r^*\right) \tag{9}$$

$$p = \left[\left(\frac{x}{y}\right)^{\kappa} e^{r^*}\right]^{-t/(1+\kappa t)}. \tag{10}$$

Proof. The formula for the reference rate is

$$r = \kappa \cdot \ln\left(\frac{X}{y}\right) + r^* = \kappa \cdot \ln\left(\frac{xe^{-rt}}{y}\right) + r^*$$
$$\Rightarrow r = \frac{1}{1+\kappa t}\left(\kappa \ln \frac{x}{y} + r^*\right).$$

Then, it is easy to see the the reference price as

$$p = \mathrm{e}^{-tr} = \exp\left[\frac{-t}{1+\kappa t}\left(\kappa \ln \frac{x}{y} + r^*\right)\right] = \left[\left(\frac{x}{y}\right)^{\kappa} e^{r^*}\right]^{\frac{-t}{1+\kappa t}}.$$

□

Now, we derive the pool reserves from state (X, y).

Lemma 2. *Let t be the time-to-maturity. Then, the bond quantity in the pool can be expressed in terms state (X, y) as*

$$x = X^{1+\kappa t}\left(\frac{e^{r^*}}{y^{\kappa}}\right)^{t}. \tag{11}$$

Proof. Using Eq. (10) of Lemma 1,

$$X = xp = x\left[\left(\frac{x}{y}\right)^{\kappa} e^{r^*}\right]^{-t/(1+\kappa t)} = x^{1/(1+\kappa t)}\left(\frac{e^{r^*}}{y^{\kappa}}\right)^{-t/(1+\kappa t)}$$
$$\Rightarrow x = \left[X\left(\frac{e^{r^*}}{y^{\kappa}}\right)^{t/(1+\kappa t)}\right]^{1+\kappa t} = X^{1+\kappa t}\left(\frac{e^{r^*}}{y^{\kappa}}\right)^{t}.$$

□

Next, we show how the pool reserves can be computed from r given an arbitrary function $R()$.

Theorem 1. *Suppose that the AMM starts at state (X_0, y_0) with initial rate r_0. Let t be the current time-to-maturity, when the AMM rate is r. Then the bond quantity x and the cash amount y in the pool are related to rate r as follows:*

$$x = x_0 \frac{1+R^{-1}(r_0)}{R^{-1}(r_0)} \frac{R^{-1}(r)}{1+R^{-1}(r)} \exp \int_{r_0}^{r} \frac{tdr}{1+R^{-1}(r)} \tag{12}$$

$$y = x_0 \frac{1+R^{-1}(r_0)}{R^{-1}(r_0)} \frac{1}{1+R^{-1}(r)} \exp \int_{r_0}^{r} \frac{tdr}{1+R^{-1}(r)}. \tag{13}$$

Proof. We have

$$r = R\left(\frac{X}{y}\right) \Rightarrow y = \frac{X}{R^{-1}(r)} = \frac{xe^{-rt}}{R^{-1}(r)}. \tag{14}$$

The reference price is the instantaneous price at time-to-maturity t, hence

$$\frac{-dy}{dx} = p = e^{-rt}.$$

Thus we have the following ODE system to find x and y,

$$\begin{cases} y = \frac{xe^{-rt}}{R^{-1}(r)} \\ \dfrac{dy}{dx} = -e^{-rt} \end{cases}$$

with initial condition (x_0, y_0, r_0). It is not difficult to solve this ODE system, so we skip the detailed steps here due to limited paper space. The formulas for x, y in Eq. (12) and Eq. (13) are precisely the solution to this ODE system.

□

As a corollary, we have the result below showing nice closed-forms for computing the pool reserves from reference rate r when R is our logarithmic rate function.

Theorem 2. *Suppose that the AMM starts at state (X_0, y_0) with initial rate r_0. Let t be the current time-to-maturity, when the AMM rate is r. Then the bond quantity x and the cash amount y in the pool are related to rate r as follows:*

$$x = X_0 e^{r_0} \cdot \left[e^{\kappa^{-1}(r-r_0)} \cdot \frac{e^{\kappa^{-1}(r_0-r^*)}+1}{e^{\kappa^{-1}(r-r^*)}+1}\right]^{t\kappa+1} \tag{15}$$

$$y = y_0 \cdot \left[\frac{e^{\kappa^{-1}(r_0-r^*)}+1}{e^{\kappa^{-1}(r-r^*)}+1}\right]^{t\kappa+1}. \tag{16}$$

Proof. The proof is obvious by applying Theorem 1 with $R(\psi) = \kappa \ln \psi + r^*$ and doing some simple derivations. □

B. Constant-Function Invariant

We have an important result below showing that, at any given time, trading on BondMM is equivalent to that on a constant-function AMM.

Theorem 3. *Let t be time-to-maturity. BondMM has a constant-function invariant,*

$$Kx^{\alpha} + y^{\alpha} = C, \tag{17}$$

where C is a constant specific to time t, i.e., constant over all transactions happening at this time, and

$$\alpha \triangleq \frac{1}{1+t\kappa}, \; K \triangleq e^{\frac{-tr^*}{1+t\kappa}} \tag{18}$$

4

The pool inventory must satisfy this equality before and after any order that is executed at time-to-maturity t.

Proof. Because $r = \kappa \ln \psi + r^*$, we have $\psi = e^{\kappa^{-1}(r-r^*)}$ and so

$$\frac{y}{x} = \frac{y}{Xe^{rt}} = \psi^{-1}e^{-rt} = e^{-\kappa^{-1}(r-r^*)-rt}$$
$$\Rightarrow \kappa^{-1}(r-r^*) = \frac{-tr^* + \ln\frac{x}{y}}{1+t\kappa}.$$

Now, let

$$C \triangleq y_0^{\frac{1}{1+t\kappa}}\left[e^{\kappa^{-1}(r_0-r^*)} + 1\right]$$

where y_0 and r_0 are the cash reserve and rate at the pool initialization. We have

$$\begin{aligned}
C^{t\kappa+1} &= y_0 \cdot \left(e^{\kappa^{-1}(r_0-r^*)}+1\right)^{t\kappa+1} \\
&= y\left(e^{\kappa^{-1}(r-r^*)}+1\right)^{t\kappa+1} \text{(thanks to Eq. (16))} \\
&= y\left(e^{\frac{-tr^*}{1+t\kappa}}\left(\frac{x}{y}\right)^{\frac{1}{1+t\kappa}}+1\right)^{t\kappa+1} \\
\Rightarrow C &= y^{\frac{1}{1+t\kappa}}\left[K\left(\frac{x}{y}\right)^{\frac{1}{1+t\kappa}}+1\right] = Kx^\alpha + y^\alpha.
\end{aligned}$$

□

Theorem 4. *Let t be time-to-maturity. In terms of AMM state, BondMM has the following constant-function invariant*

$$y^\alpha\left(\frac{X}{y}+1\right) = C. \tag{19}$$

Proof. This is a corollary of Theorem 3. Expressing $x = Xe^{rt}$ and applying some simple derivations, we obtain this constant-function property in terms of state (X, y). □

C. Order Execution

Suppose that the current time-to-maturity is t when BondMM is at state (X, y) with rate r and an order $(\Delta x, \Delta y)$ arrives. By using the constant-function property, we show below how to calculate Δy from Δx. The other direction, computing Δx from Δy, can similarly be derived.

Given X, we compute the corresponding bond quantity in the pool, $x = Xe^{rt}$. Then, according to Theorem 3, because

$$K(x+\Delta x)^\alpha + (y+\Delta y)^\alpha = C = Kx^\alpha + y^\alpha,$$

we can express Δy in terms of Δx,

$$\begin{aligned}
\Delta y &= \left[C - K(x+\Delta x)^\alpha\right]^{1/\alpha} - y \\
&= \left(Kx^\alpha + y^\alpha - K(x+\Delta x)^\alpha\right)^{1/\alpha} - y.
\end{aligned} \tag{20}$$

We can now calculate the price of an order given the state.

Theorem 5. *The average price for executing an order of size Δx at time-to-maturity t can be expressed in terms of current state (X, y),*

$$\overline{p} = \frac{y}{\Delta x}\left[1 - \left(\frac{X}{y} + 1 - \left(\left(\frac{X}{y}\right)^{1/\alpha} + \frac{K^{1/\alpha}\Delta x}{y}\right)^\alpha\right)^{1/\alpha}\right] \tag{21}$$

where α, K are defined in Theorem 3.

Proof. From Eq. (11) in Lemma 2, we have

$$K^{1/\alpha}x = K^{1/\alpha}X^{1+\kappa t}\left(\frac{e^{r^*}}{y^\kappa}\right)^t = X^{1/\alpha}y^{-\kappa t}.$$

Eq. (20) then becomes

$$\begin{aligned}
\Delta y &= \left(Kx^\alpha + y^\alpha - K(x+\Delta x)^\alpha\right)^{1/\alpha} - y \\
&= \left(Xy^{-\kappa t\alpha} + y^\alpha - (X^{1/\alpha}y^{-\kappa t} + e^{-r^*t}\Delta x)^\alpha\right)^{1/\alpha} - y \\
&= y\left(\frac{X}{y} + 1 - \left(\left(\frac{X}{y}\right)^{1/\alpha} + e^{-r^*t}\frac{\Delta x}{y}\right)^\alpha\right)^{1/\alpha} - y.
\end{aligned}$$

Putting this in $\overline{p} = \frac{-\Delta y}{\Delta x}$, we get the formula needed to prove. □

D. Liquidity Provisioning and Financial Stability

An LP can contribute liquidity anytime by adding a cash amount to the pool. Let this amount be $\Delta y > 0$. As a result, the AMM moves state from (X, y) to $(X^{new}, y^{new} = y + \Delta y)$. It is required that the reference rate be unchanged. This is equivalent to keeping the same bond-to-cash ratio. That is,

$$\psi^{new} = \frac{X^{new}}{y^{new}} = \frac{X^{new}}{y+\Delta y} = \psi = \frac{X}{y}.$$

Hence,

$$X^{new} = \frac{X(y+\Delta y)}{y}.$$

The new LP will own a fraction $s = \frac{\Delta y}{E+\Delta y}$ of the pool where E is the present equity in the pool (to be describe below). The other LPs will have their share diluted accordingly.

When withdrawing liquidity, that LP can withdraw a fraction $s_1 \leq s$ of the pool at the most. If the state is (X, y), the cash amount withdrawn will be $s_1 y$. The new state will be

$$X^{new} = X(1-s_1),\ y^{new} = y(1-s_1)$$

to keep the reference rate and bond-to-cash ratio unchanged. The other LPs will have their share diluted accordingly.

Financial stability. At any point of time, the liquidity pool should have sufficient cash to fulfill orders that sell bonds to the pool or that redeem bonds at redemption. Therefore, for implementation in practice, when the LPs withdraw liquidity or a new lending position is being created, i.e., bond buying, we should only allow this action if the AMM is able to maintain its equity above a certain threshold.

Let b and l be the number of active bonds minted by borrowers and that currently held by lenders, respectively. The

5

net equity at the current time is $E = y + (b - l)p$ which is the cash reserve plus the net cash that would materialize if all debt positions (bond positions) were closed now. Since the business model for the AMM is to generate revenue mainly from transaction fees, equity E should not necessarily be higher than initial cash liquidity y_0. This is because a higher equity than y_0 means more trading costs, thus unattractive to the traders. On the other hand, a lower equity is not only unattractive to the LPs but also makes the AMM vulnerable to liquidity and solvency risks. So in the case that equity is trending substantially below y_0, we should discourage LP withdrawal and bond buying. In an equilibrium, E should equal y_0. In other words, a desirable AMM is one that keeps E as stable and as close to y_0 as possible.

IV. Numerical Results

We conducted an evaluation with both real-world and synthetic datasets to investigate and validate key properties of BondMM. Two maturity durations were considered: $T = 1$ year and $T = 10$ years. Numerical results are discussed with respect to four important aspects: 1) Interest rate, 2) Price impact, 3) Equity, and 4) Capital efficiency.

Comparison benchmarks. We compare BondMM to Yield and Notional, today's leading fixed-rate AMM protocols. They would start with the same initial cash $y_0 = 1$ DAI and initial rate r_0 (which depends on the input dataset, explained later), and perform under the same sequence of input orders. BondMM and Notional use the same setting for $r^* = r_0$ and $\kappa = 0.02$. The initial bond quantity x_0 in each protocol can be different depending on the equation that links the pool reserves to the interest rate. Our simulation assumes zero transaction fee. Any fee mechanism can be orthogonally implemented on top of these protocols with the same effect.

Transaction types. There are four activities: borrowing, lending, closing a borrowing position, and closing a lending position. We randomize them with equal probability. Precisely, a sell order is treated as either a new borrowing position or a lending closing position with equal probability $1/2$. Similarly, a buy order is treated as either a new lending position or a borrowing closing position with equal probability $1/2$. We assume no LP change during the simulation run.

Datasets. We evaluated with one real-world and two synthetic datasets, illustrated in Figure 1.

- Real-world (AAVE): We collected a time series of borrowing stable rates from the ETH lending pool on AAVE Protocol with daily frequency over one year from Dec 1st, 2022 to Nov 30, 2023. This data set represents a market scenario in which the market rates are trending in one direction towards more lending over the time.
- Synthetic (CIR and VASICEK): We generated two synthetic datasets using the Cox–Ingersoll–Ross (CIR) [22] and VASICEK model [23] with realistic statistics. These are the two most popular models in financial mathematics to generate interest rates over time. The CIR dataset has 525,600 transactions every minute over 1 year, representing a market with a wider rate variation. The VASICEK dataset has 5,256,000 transactions every minute over 10 years, representing a normal market scenario where rates are mean-reversing, up and down normally.

For comparing the protocols under the same sequence of orders, to generate a realistic order input, our approach is to use the above market rate data. Given a time series of market rates (r_t), from the AAVE, CIR, or VASICEK dataset, we reconstruct a corresponding time series of orders (Δx_t) such that executing these orders on an independent interest rate protocol, which we dub "Linear Protocol", would result in these market rates (r_t). We then apply these same orders (Δx_t) on BondMM, Yield, and Notional to compare. The Linear Protocol is described by the following equation, $r_t = r_{\min} + (r_{\max} - r_{\min}) \ln \frac{x_t}{y_t}$, where parameters $r_{\min}, r_{\max}$ are chosen symmetrically around initial rate r_0.

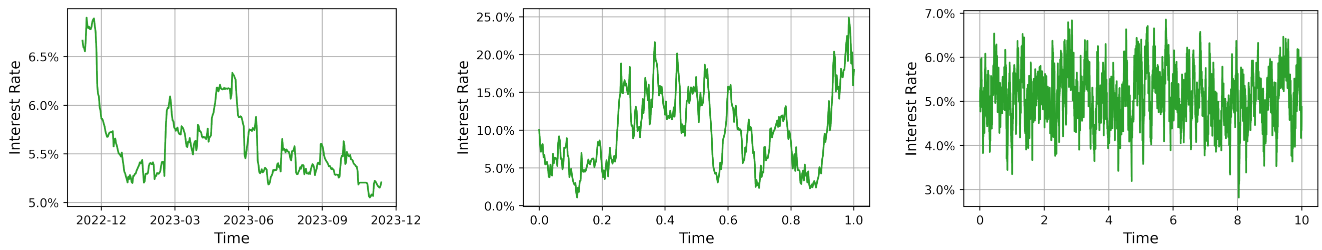
A. Interest Rate

We plot the interest rates over the time of the protocols as a result of processing the input orders in Figure 2 for different simulation settings: AAVE dataset with 1-year maturity, CIR dataset with 1-year maturity, and VASICEK dataset with 10-year maturity. In all scenarios, it is easy to see that the interest rate curve of Yield is the most volatile one, which oftentimes reaches negative rates. In contrast, Notional and BondMM are much more stable and have similar rate volatility. Especially, extreme volatility is observed for Yield in the case 1-year maturity for both AAVE dataset (Figure 2(a)), where the rate can go negative reaching -150%, and CIR dataset (Figure 2(b)) where the rate can reach -100%. These rates are absolutely surreal. Even for the VASICEK dataset which represents a stable market, Yield rates still fluctuate widely and can go below zero (see Figure 2(c)). It is not desirable to have a lending protocol with extremely volatile rates, and, in this aspect, Yield is not practical. Both BondMM and Notional offer stable rates because they can choose an arbitrarily small κ value to keep price/rate volatility under control. In contrast, Yield does not have this flexibility.

B. Price Impact

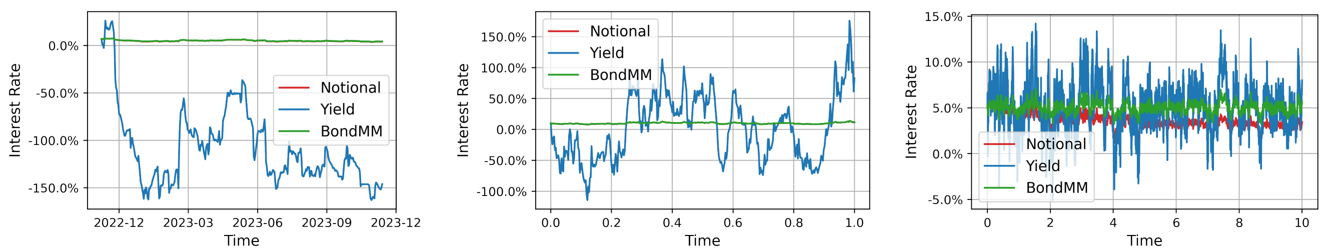
Traders do not like price slippage when executing a large order. Worse slippage means higher trading cost. To compare the price impact of the three protocols, we computed the trade price of each order as a percentage slippage from the current price. Applying to the different datasets and different maturities, Figure 3 shows the price-impact histogram, plotting the number of orders that experience a given price slippage. As seen, Yield is extremely expensive. Its histogram is almost flat over all price impact values, implying a very high degree of price slippage (the average value is so high that it is not shown in the plot). Notional is not that extreme, but substantially more expensive than BondMM, especially with longer maturity. Let us look at the 95%-quantile worst-case scenario. With 1-year maturity, Notional has a price impact about 1.5 times higher than BondMM; 0.17% vs. 0.12% for AAVE dataset (Figure 3(a)) and 0.04% vs. 0.027% for CIR (Figure 3(b)) . The gap is wider for the 10-year maturity, as

6



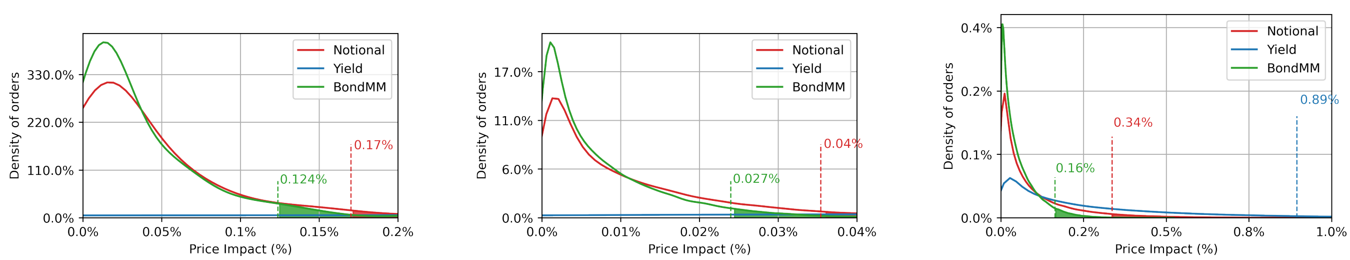
(a) AAVE (365 tx, daily frequency) (b) CIR (500K+ tx, minute frequency) (c) VASICEK (5M+ tx, minute frequency)

Fig. 1: Market rate datasets used in the experiments: (a) Real-world ETH stable rates from AAVE protocol; (b) Synthetic rates generated by the CIR model; and (c) Synthetic rates generated by the VASICEK model.



(a) AAVE: 1-year maturity (b) CIR: 1-year maturity (c) VASICEK: 10-year maturity

Fig. 2: Interest rate of each protocol as a result of processing input orders over the time. In plots (a) and (b), the Notional and BondMM curves are highly similar, hence seen as the same.



(a) AAVE: 1-year maturity (b) CIR: 1-year maturity (c) VASICEK: 10-year maturity

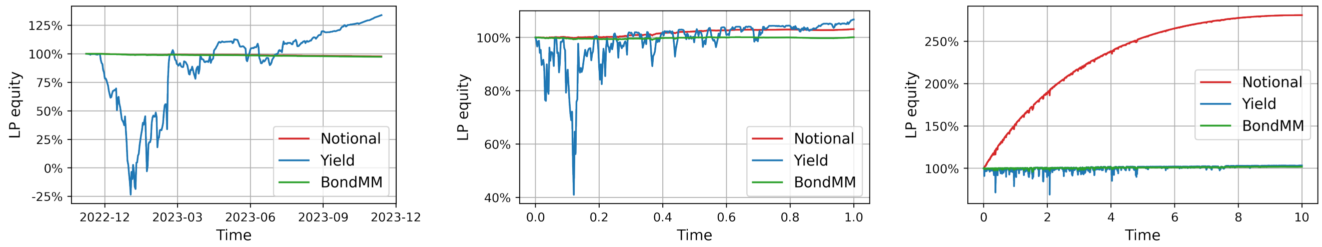
Fig. 3: Price impact measured as a percentage slippage from current price when pricing an order. The y-axis is the proportion of orders having a given slippage. In plots (a) and (b), the Yield impact is very small, hence seen close to the y = 0% line.

seen in Figure 3(c), where Notional (0.34%) is more than twice costlier than BondMM (0.16%) and Yield (0.89%) is 5+ times costlier. This evaluation obviously shows that BondMM pricing is the most attractive for traders.

C. Equity

A fair bond protocol needs equity to be as close to the initial equity as possible. Another important criterion is the stability of the equity PnL: the more stable it is, the better LPs are protected against uncertainty of the interest market. Figure 4(a) (for the AAVE dataset with 1-year maturity), Figure 4(b) (for the CIR dataset with 1-year maturity), and Figure 4(c) (for VASICEK dataset with 10-year maturity) show the LP equity of each protocol over the time. In all three scenarios, BondMM is the absolute winner because its equity is not only the closest to initial equity but also is the most stable. For the short maturity (1-year), BondMM and Notional have comparable equity. This is understandable because the bond maturity is too short for these two protocols to substantially differentiate from each other. However, Yield's equity curve is extremely bad, particularly during the early phase of the protocol. In the AAVE scenario, Yield's equity can decrease below zero, which is unacceptable. In the CIR scenario, the LPs can lose as much as 60% of the initial equity. We see that this equity drop happens when the rate also goes negative, seeing a lot of bond buying (lending). The results here point out a critical weakness of Yield: it would potentially collapse in a lending-heavy market. In contrast, BondMM and Notional still maintains stable equity close to the initial.

7



(a) AAVE: 1-year maturity (b) CIR: 1-year maturity (c) VASICEK: 10-year maturity

Fig. 4: LP equity over the time. In plot (a), the Notional and BondMM curves are highly similar, hence seen as the same.

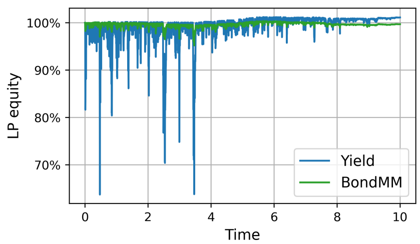


Fig. 5: LP equity over the time: BondMM versus Yield for the VASICEK dataset with 10-year maturity.

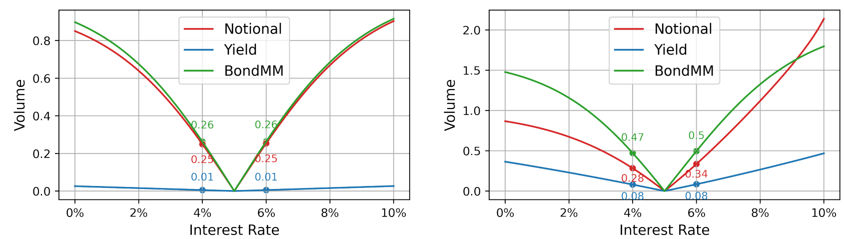


Fig. 6: Capital efficiency: The y-axis is the bond volume, relative with the bond quantity in the pool, that can be fulfilled to reach a given rate. Here, (left) 1-year maturity; and (right) 10-year maturity.

In the more stable market with the VASICEK dataset (Figure 4(c)), where the rates are mean-reversed and the maturity is longer (10 years), the more interesting observation is about Notional. Due to its greedy trade pricing, as the market is stable with balanced net bond positions, Notional accumulates a lot more cash from trading compared to the other protocols, explaining why its equity grows higher noticeably. This is not good though. Notional quickly doubles equity after two years. This is too greedy and unrealistic; no traders would come to Notional. As aforementioned, we want an equity that is stable and as close to the initial equity as possible. We see that BondMM satisfies both conditions. Yield is still unstable during early time, when it can lose 30% of equity; see Figure 5 for a zoom-in plot comparing BondMM and Yield.

D. Capital Efficiency

While equity is a factor to represent fairness between the LP and the traders, capital efficiency affects LP exclusively. It is always desirable to maximize it. To compare capital efficiency of the three protocols, we assume that they reach an equilibrium state having the same cash reserve in the pool and the the same rate as market rate, and then compute how much bond volume can be fulfilled by each protocol to change the rate by the same amount. The larger this volume, the more efficient the LP capital is utilized. In other words, we have a better liquidity depth. Specifically, we set the market rate at 5% and the results are shown in Figure 6 for two cases of maturity, $T = 1$ year and $T = 10$ years. Yield is obviously the least capital efficient. BondMM and Notional have similar capital efficiency with 1-year maturity. With the longer 10-year maturity, BondMM is noticeably better. For example, to move the current rate from 5% to 4%, BondMM can accommodate a trade volume 0.47 (in bond unit), where Notional and Yield can only do 0.28 and 0.08. Put another way, BondMM is almost twice more capital efficient than Notional and 50 times more than Yield. This study confirms our hypothesis that our rate formulation results in a better liquidity concentration around realistic rate ranges (in this evaluation, the 5% market rate), especially when the maturity duration increases.

V. Conclusions

We have proposed BondMM, a new AMM protocol for fix-rate lending and trading. It is better than today's fixed-rate AMMs in terms of price impact (more attractive to users), capital efficiency (more attractive to LPs), and equity stability (better balance between trading costs to users and equity for LPs to avoid liquidity and solvency risks). Also, in terms of algorithm construction, existing solutions remain too simplistic or engineering-driven, lacking verifiable mathematical and financial justifications. As such, to be financially safe, their applicability is limited to short-maturity products only. In contrast, BondMM can work for both short and long maturities. It is a constant-function AMM constructed with elegant closed-form mathematics. Its rate and price formulations satisfy the desirable path-independence property. One issue we have not mentioned in this paper is to investigate the effect of arbitrage activities between the AMM and the outside market. This will be our future work, in which we will also introduce fee mechanisms to compensate for LP loss due to arbitrage as well as other types such as loss-versus-rebalancing loss.

8

REFERENCES

[1] D. A. Tran, M. T. Thai, and B. Krishnamachari, Eds., *Handbook on Blockchain*. Springer, 2022.

[2] BarnBridge, "Barnbridge white paper," 2021. [Online]. Available: https://github.com/BarnBridge/BarnBridge-Whitepaper

[3] Swivel Finance. [Online]. Available: https://swivel.finance/

[4] AAVE, "AAVE protocol (white paper)," 2020. [Online]. Available: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf

[5] R. Leshner and G. Hayes, "Compound: The money market protocol (white paper)," 2019. [Online]. Available: https://compound.finance/documents/Compound.Whitepaper.pdf

[6] A. Niemerg, D. Robinson, and L. Livnev, "Yieldspace: An automated liquidity provider for fixed yield tokens (white paper)," 2020. [Online]. Available: https://yield.is/YieldSpace.pdf

[7] Notional Finance, "Notional protocol (white paper)," 2020. [Online]. Available: https://docs.notional.finance/developers/whitepaper/whitepaper

[8] Pendle Finance. [Online]. Available: https://www.pendle.finance/

[9] Term Finance. [Online]. Available: https://term.finance/

[10] Element Finance, "The element protocol construction paper." [Online]. Available: https://paper.element.fi/

[11] J. C. Schlegel, M. Kwaśnicki, and A. Mamageishvili, "Axioms for constant function market makers," in *Proceedings of the 24th ACM Conference on Economics and Computation*, ser. EC '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1079. [Online]. Available: https://doi.org/10.1145/3580507.3597720

[12] G. Angeris, A. Evans, T. Chitra, and S. P. Boyd, "Optimal routing for constant function market makers," in *EC '22: The 23rd ACM Conference on Economics and Computation, Boulder, CO, USA, July 11 - 15, 2022*, D. M. Pennock, I. Segal, and S. Seuken, Eds. ACM, 2022, pp. 115–128. [Online]. Available: https://doi.org/10.1145/3490486.3538336

[13] G. Angeris, A. Agrawal, A. Evans, T. Chitra, and S. Boyd, *Handbook on Blockchain*. Springer, 2022, ch. Constant Function Market Makers: Multi-asset Trades via Convex Optimization, pp. 415–444.

[14] V. Buterin, "On path independence," 2017. [Online]. Available: https://vitalik.ca/general/2017/06/22/marketmakers.html

[15] MakerDAO, "The maker protocol: Makerdao's multi-collateral dai (mcd) system (white paper)," 2019. [Online]. Available: https://makerdao.com/en/whitepaper

[16] H. Adams, N. Zinsmeister, and D. Robinson, "Uniswap v2 core," 2020. [Online]. Available: https://uniswap.org/whitepaper.pdf

[17] F. Martinelli and N. Mushegian, "Stableswap - efficient mechanism for stablecoin liquidity," 2019. [Online]. Available: https://balancer.fi/whitepaper.pdf

[18] Spectra Finance. [Online]. Available: https://www.spectra.finance/

[19] H. Adams, N. Zinsmeister, M. Salem, R. Keefer, and D. Robinson, "Uniswap v3 core," 2021. [Online]. Available: https://whitepaper.io/document/52/bancor-whitepaper/

[20] V. Nguyen, "A study on amm for trading fixed yield and pendle's v2 principal token amm," October 2022. [Online]. Available: https://github.com/pendle-finance/pendle-v2-resources/blob/main/whitepapers/V2_AMM.pdf

[21] A. Othman, D. M. Pennock, D. M. Reeves, and T. Sandholm, "A practical liquidity-sensitive automated market maker," *ACM Trans. Econ. Comput.*, vol. 1, no. 3, sep 2013. [Online]. Available: https://doi.org/10.1145/2509413.2509414

[22] J. C. Cox, J. E. Ingersoll, and S. A. Ross, "A theory of the term structure of interest rates," *Econometrica*, vol. 53, no. 2, pp. 385–407, 1985. [Online]. Available: http://www.jstor.org/stable/1911242

[23] O. Vasicek, "An equilibrium characterization of the term structure," *Journal of Financial Economics*, vol. 5, no. 2, pp. 177–188, 1977. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0304405X77900162

9