ChickenBonds: Self-Bootstrapping Treasury

Liquity Team

April 5, 2021

Abstract

ChickenBonds is an autonomous self-bootstrapping treasury system that acquires yield-bearing tokens (TKN) through a novel bonding mechanism: users may bond TKN and start accruing a "boosted" derivative token (bTKN) in return.

At any time, bond holders can either retrieve their principal foregoing the accrued amount ("chicken out") or trade it in against the accrued bTKN ("chicken in").

The TKN acquired by the system backs the bTKN supply: while a portion of the TKN in the treasury is directly redeemable by bTKN holders, another portion is contributing permanently to the redeemable amount through the generated yield.

By retaining the revenue from the entire treasury including outstanding bonds, the system amplifies the redemption value of the bTKN, resulting in a rising price floor. This yield amplification makes it attractive to buy and hold bTKN in addition to obtaining it through bonding.

ChickenBonds are versatile and can be used by protocols to obtain DEX liquidity for their token at no cost or to pursue more sophisticated liquidity management strategies. An interesting use case involves bonding LP tokens.

1 Introduction

TODO

2 Treasury

The protocol operates a *Treasury R* consisting of three logical parts ("buckets"): *Pending Bucket*, *Acquired Bucket* and *Permanent Bucket*.

Each bucket contains a certain quantity of TKN or other assets whose value can be readily expressed in TKN. We suppose that the assets in the buckets can be invested in a third-party protocol (e.g. staked natively or deposited to a DEX) to generate yield while being withdrawable at any time.

The current state of the Treasury can thus be described by the following tuple:

$$R := (q_p, q_a, q_d, r_p, r_a, r_d) \tag{1}$$

where q_p, q_a and q_d stand for the quantities held by the respective buckets, and r_p, r_a, r_d for their rates of return. Note that the buckets as logical quantities do not need to correspond to the physical investment vehicles. A bucket may invest its funds to multiple investment vehicles, and several buckets may use the same venue earning the same rate of return.

The buckets are used as follows:

- *Pending Bucket.* Contains the TKN bonded by users that are still active bond holders, i.e. whose deposits haven't been acquired by the protocol yet. Since bond holders may withdraw their bonded TKN at will (see 3.1.1 "chicken out"), their deposits are considered as pending. The yield earned by the Pending Bucket is credited to the Acquired Bucket.

- *Acquired Bucket.* Contains a portion of the TKN relinquished by former bond holders after a "chicken in" event (see 3.1.2) as well as the yield of the Pending and the Permanent Buckets. The Acquired Bucket directly backs the bTKN supply through the redemption mechanism (see 3.2).

- *Permanent Bucket.* Contains the other portion of the TKN relinquished by former bond holders. The yield earned by the Permanent Bucket is credited to the Acquired Bucket.

Having a Permanent Bucket may turn out as unnecessary or even detrimental for some use cases. As an alternative, the protocol can thus be set up without a Permanent Bucket (see 5.2).

2.1 Yield Amplification and Reinvestment

Given that the Acquired Bucket earns a return not only on its own assets, but additionally receives the returns from the Pending and the Permanent Buckets, it achieves an amplified return $r_a^\star > r_a$ on the amount q_a. We call this useful property *yield amplification*.

As the protocol doesn't distribute any yields, but reinvests them inside the Acquired Bucket, the value of the Acquired Bucket will grow over time even without the inflow of new capital. Due to the yield amplification, the Acquired Bucket grows faster than if the same funds were invested regularly with a (compounding) rate of return r_a.

2.2 Acquired Bucket backing the Boosted Token

The protocol maintains a supply S of a *Boosted Token* (bTKN) according to preset rules for minting and burning. The bTKN supply is directly backed by the Acquired Bucket, i.e. it can be redeemed against a pro rata share of the assets (usually TKN) held therein (see 3.1.1).

We call the amount of TKN for which each bTKN can be redeemed for the *redemption price* p_r. The redemption price corresponds to the *backing ratio* and is defined as $\frac{q_a}{S}$. It is subject to the following invariant:

Invariant (Redemption price never decreases)**.** *The protocol ensures that the redemption price p_r (backing ratio) can only ever increase, but never decrease.*

3 User interactions

Users can interact with the protocol in the followings ways:

- *Bonding.* Users can bond TKN in order to earn bTKN.

- *Redemption.* Users can redeem bTKN for TKN.

3.1 Bonding

Bonding is the main use case that allows the system to build up its treasury by incentivizing users through the issuance of a *Boosted Token*, the bTKN.

Users can bond any amount b of TKN at any time in exchange for a position called *Chicken Bond*, represented by an NFT. The bonded TKN is added to the *Pending Bucket* where it earns a yield for the system's Treasury.

A Chicken Bond accrues a virtual balance $s(t)$ of bTKN over time, according to a curve that asymptotically approaches a "cap" ensuring the invariant of a never decreasing redemption price.

The accrual curve can be of the form

$$s(t) := \frac{b}{p_r} \cdot \frac{t}{t + \alpha} \tag{2}$$

α parametrizes the slope of the curve and could be automatically adjusted by the protocol in order to control the speed of the value accrual which depends on the evolving price premium (see 5.3).

We call the fraction $\frac{b}{p_r}$ the cap c which corresponds to the amount of bTKN that could be minted by the protocol such that the redemption price would be kept constant if b was entirely added to the Acquired Bucket. Therefore, the ratio between the cap and the bond corresponds to the ratio between the bTKN supply and the Acquired Bucket. Thus, we have

$$\frac{c}{b} = \frac{S}{q_a} \tag{3}$$

The owner of a *Chicken Bond* can exit their position any time by choosing either of the following options:

- *Chicken out.* Retrieve the principal foregoing the accrued bTKN.

- *Chicken in.* Obtain the accrued bTKN foregoing the bonded TKN.

In both cases, the Chicken Bond NFT is burned as the bond holder's position is closed. Depending on the option chosen, the bonded TKN is either fully paid back to the user or it transitions from the Pending Bucket to the Acquired Bucket and the Permanent Bucket according to a split calculated by the system (3.1.2).

3.1.1 Chicken out

By chickening out, the bond holder gets the entire bonded TKN back, foregoing the accrued balance of bTKN (which doesn't get minted). The option to withdraw the principal at any time makes bonding an essentially risk-free investment where the user only incurs the opportunity costs besides smart contract risks.

Upon a chicken out event at time $t + 1$, the Treasury changes as follows:

$$q_p(t + 1) := q_p(t) - b \tag{4}$$

As an alternative, the bond holder may sell their NFT on the secondary market. The buyer of the NFT has the same rights against the protocol as the original bond holder.

3.1.2 Chicken in

A user that chickens in loses the bonded TKN in exchange for the accrued balance of bTKN that is minted and paid out by the protocol. A *payout tax* may be charged on the accrued balance (reducing the user's payout) in order to incentivize liquidity for a bTKN/TKN exchange pair (see 5.5).

Instead of keeping the received bTKN, the user may sell it on the open market and opt to reinvest the proceeds by creating a new, typically larger bond (aka as "rebonding", see below).

Upon a chicken in, the protocol acquires the bonded TKN which is moved from the Pending Bucket to the Acquired Bucket and the Permanent Bucket. To that end, the bonded amount b is first split into two amounts b_a and b_d in proportion to the ratio of the currently accrued bTKN s to the cap c:

$$b_a = \frac{s}{c} \cdot b = s \cdot p_r \tag{5}$$

$$b_d = \frac{c - s}{c} \cdot b = b - s \cdot p_r \tag{6}$$

The two amounts are then added to the Acquired and the Permanent Bucket, so that the Treasury transitions to the following state:

$$q_a(t + 1) := q_a(t) + b_a = q_a(t) + s \cdot p_r \tag{7}$$

$$q_d(t + 1) := q_d(t) + b_d = q_d(t) + b - s \cdot p_r \tag{8}$$

$$q_p(t + 1) := q_p(t) - b \tag{9}$$

3.2 Redemption

At any time, a holder can redeem n bTKN for $\frac{n}{S} \cdot q_a = n \cdot p_r$ TKN. The TKN are removed from the Acquired Bucket and paid out the user, while the redeemed bTKN are burned.

The state of the Treasury changes as follows:

$$q_a(t+1) := q_a(t) \cdot (1 - \frac{n}{S}) \tag{10}$$

$$S(t+1) := S(t) - n \tag{11}$$

To throttle the rate of redemption or counter extreme situations and black swan events, the system could charge a redemption fee (see 5.4).

4 Economics of ChickenBonds

The Boosted Token bTKN derives its value from the Acquired Bucket, as its direct backing, and the yield generated by the entire Treasury.

Since the bTKN supply is redeemable for a proportional share of the acquired TKN, the market cap of bTKN should normally be worth at least the amount of TKN kept in the Acquired Bucket. This means that the redemption price p_r will act as a price floor, below which arbitrage becomes possible: people can buy bTKN for less than what they get upon redemption. Should bTKN ever drop below the redemption price, we can expect it to bounce back quickly due to the buying pressure stemming from redemptions.

In practice, we expect bTKN to trade significantly above its price floor most of the time due to the yield amplification: if we suppose that r_a corresponds to the market rate (or natural rate) r_m, the Acquired Bucket and thus p_r will grow at a higher rate than r_m given the extra yield generated by the Pending and the Permanent Bucket. This increased return warrants a price premium since the market should price in the amplified growth rate by paying a higher price for bTKN than the redemption price. We call this expected market price the *fair price* p_f and let $\lambda = \frac{p_f}{p_r}$ denote the *relative price premium* or simply *premium*, i.e. the fraction between the fair price and the redemption price.

4.1 Estimating the fair price p_f

4.1.1 Naive approach

We start by modeling a yield-bearing token TKN as an investment that pays out a future sum of money or stream of cash flows. Given a *natural rate r_m* (aka market rate or discount rate) this allows to calculate its present value (PV). One can then consider a compounding wrapper contract around TKN that retains and compounds the generated yield, giving the investor the option to redeem their wrapper share against a pro rata share of value held inside the contract. We assume that similarly to a mutual accumulation fund, the value of the investor's share in the compounding contract would reflect the inner value of the share, i.e. the value the investor receives upon redemption.

In ChickenBonds, the Acquired Bucket corresponds to a compounding wrapper with the extra benefit of receiving the return from the Pending and the Permanent Bucket in addition to its own yield. If those two other buckets were empty, the Acquired Bucket's PV expressed in TKN would be equal to its inner value, i.e. q_a, and the fair price of bTKN would correspond to its redemption price p_r. However, with non-empty Permanent and Pending Buckets acting as extra yield sources, the value of the bTKN should exceed p_r since holding bTKN is a better investment than being an investor in a simple compounding contract holding TKN. The question is by how much.

Generally, if a token x receives cash flows $F_1, F_2 ..., F_n$ from multiple different sources, the PV of the token is given by the sum of the PVs of the cash flows, i.e. $PV(x) = PV(F_1) + PV(F_2) + ... + PV(F_n)$. We suppose that the same rationale holds in principle if the yields are compounded in a redeemable wrapper contract instead of being paid out.

While this applies to the Acquired Bucket, the Permanent Bucket as such is not subject to redemption, but only its yield becomes redeemable as part of the Acquired Bucket. Therefore, we posit that the same amount of TKN inside the Permanent Bucket has a lesser impact on the price of

bTKN than it would have inside the Acquired Bucket, even when earning the same yield. We thus have to discount its price impact by some factor $d_f < 1$ to account for the permanent locking of the principal. Supposing that $r_a = r_m$ and $r_d < r_a$, and neglecting the Pending Bucket for now, we can state the present value of 1 unit of bTKN as

$$p_f = \frac{PV(AcquiredBucket) + PV(PermanentBucket)}{S} \tag{12}$$

and express the PV of the Permanent Bucket in terms of the Acquired Bucket:

$$p_f = \frac{PV(AcquiredBucket) + \frac{q_d}{q_a} \cdot \frac{r_d}{r_a} \cdot d_f \cdot PV(AcquiredBucket)}{S} \tag{13}$$

Recalling that $PV(AcquiredBucket) = q_a$, we can derive that

$$p_f = \frac{q_a + q_d \cdot \frac{r_d}{r_a} \cdot d_f}{S} \tag{14}$$

Regarding the Pending Bucket, we know that its content will eventually transition into the two other buckets (chicken in) or leave the system (chicken out). On the other hand, bonders have an incentive to rebond after chickening in (see TODO), increasing the Pending Bucket shortly after. If we assume for simplicity that the two effects cancel out in aggregate and suppose $r_p = r_a$, we can extend the fair price formula to:

$$p_f = \frac{q_a + q_p + q_d \cdot \frac{r_d}{r_a} \cdot d_f}{S} \tag{15}$$

We call this the *naive formula* since it's overly simplified, glossing over the fact that rebonding may have a different impact on the price than assumed.

4.1.2 Recursive approach

Instead of assuming that the effects of chicken ins and rebonding cancel out, we can try to estimate the future changes to the buckets based on historic user behavior or expected optimal behavior.

For changes to the TKN quantities inside the buckets, we use the notation Δq_a, Δq_p and Δq_d and let P_{In} denote the probability that a bond holder chickens in, and P_{Out} that the holder chickens out.

This allows us to express the fair price p_f as follows:

$$p_f = \frac{q_b + q_p(1 - P_{In} - P_{Out}) + (q_a + P_{In}\Delta q_a) + (q_d + P_{In}\Delta q_d)\frac{r_d}{r_m}d_d}{S_{sLQTY}\left(1 + P_{In}\Delta q_a \frac{1}{q_a}\right)} \tag{16}$$

Assumptions:

1. $P_{In} + P_{Out} \leq 1$

2. $P_{Out} = 0$, as long as bonding is profitable

3. $P_{Out} = 1$ if bonding is not profitable

We further assume for simplicty that bonding activity is more or less equally distributed over time and that bonders who got past their optimal rebonding time chicken in with $P_{In} = 1$, while nobody chickens in before reaching that point. This way, we can split the group of bonders into two segments, the ones before and the ones after the rebonding

We model P_{In} by setting it to the inverse of the optimal rebonding time.

$$P_{In} = \min\left(1, \frac{1}{T_{opt}}\right) \tag{17}$$

Bucket changes based on history We can estimate Δq_a and Δq_d based on the historic average, assuming that users would rebond when reaching the same fraction of their cap as in the past:

$$\Delta q_a = q_p \frac{q_a}{q_a + q_d} \tag{18}$$

$$\Delta q_d = q_p \frac{q_d}{q_a + q_d} \tag{19}$$

There should also exist an historical average rebonding period $\overline{T}$ corresponding to the current relation between q_a and q_d:

$$\frac{q_a}{q_a + q_d} = \frac{\overline{T}}{\overline{T} + 1} \tag{20}$$

$$\overline{T} = \frac{q_a}{q_d} \tag{21}$$

Even though $\overline{T}$ may not necessarily reflect the optimal rebonding strategy, we can use it to estimate P_{In}:

$$P_{In} = \frac{1}{\overline{T}} = \frac{q_d}{q_a} \tag{22}$$

This leads to following fair price (where q_b stands for the rebonded amount):

$$p_f = \frac{q_b + q_p \left(1 - \frac{q_d}{q_a}\right) + \left(q_a + \frac{q_d}{q_a} q_p \frac{q_a}{q_a+q_d}\right) + \left(q_d + \frac{q_d}{q_a} q_p \frac{q_d}{q_a+q_d}\right) \frac{r_d}{r_m} d_d}{S_{sLQTY} \left(1 + \frac{q_d}{q_a} q_p \frac{q_a}{q_a+q_d} \frac{1}{q_a}\right)} \tag{23}$$

$$p_f = \frac{q_b + q_p \left(1 - \frac{q_d}{q_a}\right) + q_a \left(1 + \frac{q_d}{q_a} \frac{q_p}{q_a+q_d}\right) + q_d \left(1 + \frac{q_d}{q_a} \frac{q_p}{q_a+q_d}\right) \frac{r_d}{r_m} d_d}{S_{sLQTY} \left(1 + \frac{q_d}{q_a} \frac{q_p}{q_a+q_d}\right)} \tag{24}$$

$$p_f = \frac{q_b + q_p \left(1 - \frac{q_d}{q_a}\right)}{S_{sLQTY} \left(1 + \frac{q_d}{q_a} \frac{q_p}{q_a+q_d}\right)} + \frac{q_a + q_d \frac{r_d}{r_m} d_d}{S_{sLQTY}} \tag{25}$$

Assuming that all rebonders can sell their bTKN at the fair price without incurring any slippage, we can express q_b as follows:

$$q_b = p_f P_{In} \Delta q_a \frac{1}{q_a} = \frac{q_d}{q_a} \frac{q_a}{q_a + q_d} \frac{p_f}{p_r} \tag{26}$$

Resulting in:

$$p_f = \frac{q_p \left(1 + \frac{q_d}{q_a+q_d} \frac{p_f}{p_r} - \frac{q_d}{q_a}\right)}{S_{sLQTY} \left(1 + \frac{q_d}{q_a} \frac{q_p}{q_a+q_d}\right)} + \frac{q_a + q_d \frac{r_d}{r_m} d_d}{S_{sLQTY}} \tag{27}$$

Bucket changes based on optimal rebonding strategy TODO

4.1.3 Yield-driven approach

We can also consider the evolution of the fair price of bTKN over a given time period and compare it with an expected yield.

Assuming rational markets and no additional risks for holding bTKN compared to holding (and staking) TKN, can we suppose that the market price of bTKN would grow at the same rate as the value of a staked TKN? The rationale being that if the market expects bTKN to grow faster in the future, it should pay a higher price for bTKN now, meaning that the initial price premium should reflect the future growth potential and cancel out any oversized yield from the start. Or is

economically possible under perfectly rational conditions that a portfolio A has a higher yield than a portfolio B with the same risk profile?

If we know what yield to expect, we could formulate a differential equation to derive a formula for p_f or at least a parameter α for the impact of the pending bucket in the naive formula:

$$p_f = \frac{q_a + \alpha q_p + q_d \cdot \frac{r_d}{r_a} \cdot d_f}{S} \tag{28}$$

We could then state the following equation:

$$\frac{p_f(t+365) - p_f(t)}{p_f(t)} = r_s = r_m \tag{29}$$

Given that people would rebond during the course of the year, it's simpler to consider a time period that corresponds to an optimal rebonding period for the current premium, which we suppose to be constant as long as no rebonding happens (all else being equal). This means, we assume away any new bonds and chicken-outs, and only consider the group of existing bonders which all rebond perfectly.

This allows us to reformulate the equation, which could be solved for α:

$$\frac{p_f(t+T_{opt}) - p_f(t)}{p_f(t)} = \frac{T_{opt}}{365} r_m \tag{30}$$

4.2 Profitability of bonding

Bonding is only profitable for $\lambda > 1$, i.e. if $p_f > p_r$. In that case, we can easily calculate the break even point which is reached when $s(t) > b$, i.e.

$$\frac{p_f}{p_r} \frac{t}{t+\alpha} > 1 \tag{31}$$

Solving for t yields:

$$t > \frac{\alpha p_r}{p_f - p_r} \tag{32}$$

We can then write the break even point in terms of λ

$$t > \frac{\alpha}{\lambda - 1} \tag{33}$$

The actual rate of return from bonding depends on λ and the shape (steepness) of the accrual curve $s(t)$ given by the parameter α.

4.3 Optimal rebonding strategy

When people bond, they get an accruing balance of bTKN that they can trade in against the bonded amount of TKN fixed at the beginning. That is, the system doesn't auto-compound the accrued profits from bonding. In order to benefit from a compounding effect, bond holders can chicken in, sell their received bTKN for TKN and reinvest the TKN to create a new, larger bond, and repeat that process over and over again.

In the following, we analyze the optimal rebonding strategy for users with a sufficiently long time horizon by deriving a time period that maximizes the profitability of (regular) rebonding for a given system configuration.

4.3.1 Optimal rebonding time

We assume b, p_r and p_f stay constant. Moreover, we suppose $p_f > p_r$ all the time, which is a prerequisite for bonding to be profitable.

TODO: Do we need to prove that rebonding in *regular* intervals is the optimal strategy? If we can show that rebonding at time $t < T$ of a bond created at time 0 yields a higher ROI than simple

bonding at time T, we could apply the same reasoning to the new bond created at t. As long as all factors determining the profitability are constant (and since the ROI is independent of the absolute bond size), this implies that rebonding of the new bond becomes more attractive after the exact same time period, thus at $2t$. Repeating this process, we can thus state that the profit-maximising strategy involves rebonding at a regular pace all else being equal.

Before deriving the optimal rebonding time, we start by comparing the final value of a user's bond (expressed in TKN) after a time period T without rebonding (denoted by f) with the final value of a bond that is rebonded at time $t < T$ (denoted by f_r).

TODO: with α?

No rebonding We multiply the accrued bTKN tokens from equation 2 by the market price which is supposed to be equal to the fair price p_f:

$$f = p_f \frac{b}{p_r} \frac{T}{T+1} \tag{34}$$

With rebonding At time t, when the user chickens in and rebonds, the new bond amount b' will be:

$$b' = p_f \frac{b}{p_r} \frac{t}{t+1} \tag{35}$$

From that point until the end there will be a period of $T - t$, where the outcome of the bond will be:

$$f_r = p_f \frac{b'}{p_r} \frac{T-t}{T-t+1} \tag{36}$$

So finally we have

$$f_r = b \left(\frac{p_f}{p_r}\right)^2 \frac{t}{t+1} \frac{T-t}{T-t+1} \tag{37}$$

Now we generalize the approach for n regular rebonding events, expressing the resulting value in TKN for a given rebonding period t with $f_n(t)$.

$$f_n(t) = b \left(\frac{p_f}{p_r}\right)^n \left(\frac{T}{T+n}\right)^{n-1} \frac{t - \frac{n}{n-1}T}{t - \frac{n}{n-1}T + 1} \tag{38}$$

At $t = T$:

$$f_n(T) = b \left(\frac{p_f}{p_r}\right)^n \left(\frac{T}{T+n}\right)^{n-1} \frac{\frac{T}{n}}{\frac{T}{n} + 1} \tag{39}$$

$$f_n(T) = b \left(\frac{p_f}{p_r} \frac{T}{T+n}\right)^n \tag{40}$$

To find the maximum value of $f_n(T)$, we rewrite it as a function of n given T and take its derivative with respect to n:

$$f_T'(n) = b \left(\frac{p_f}{p_r} \frac{T}{T+n}\right)^n \left(\ln\left(\frac{p_f}{p_r} \frac{T}{T+n}\right) - n \frac{\frac{p_f}{p_r} \frac{T}{(T+n)^2}}{\frac{p_f}{p_r} \frac{T}{T+n}}\right) = b \left(\frac{p_f}{p_r} \frac{T}{T+n}\right)^n \ln\left(\frac{p_f}{p_r} \frac{T}{T+n}\right) - \frac{n}{T+n}$$

Then equating it to zero:

$$f_T'(n) = 0 \iff \ln\left(\frac{p_f}{p_r} \frac{T}{T+n}\right) = \frac{n}{T+n}$$

$$\frac{p_f}{p_r} \frac{T}{T+n} = e^{\frac{n}{T+n}}$$

Dividing by e both sides:

$$\frac{p_f}{re}\frac{T}{T+n} = e^{\frac{n}{T+n}-1}$$

$$\frac{p_f}{re}\frac{T}{T+n} = e^{\frac{-T}{T+n}}$$

$$\frac{T}{T+n}e^{\frac{T}{T+n}} = \frac{p_f}{re}$$

And then, it has the form $ze^z = w$ from Lambert W function: https://en.wikipedia.org/wiki/Lambert_W_function, so:

$$\frac{T}{T+n} = W\left(\frac{p_f}{re}\right)$$

$$1 + \frac{n}{T} = \frac{1}{W\left(\frac{p_f}{re}\right)}$$

$$n = \frac{T}{W\left(\frac{p_f}{re}\right)} - 1$$

To find the optimal rebonding time period T_{opt}, we divide the total considered period T by the optimal number of rebonding events n, noting that T an T is thereby cancelled out.

$$T_{opt} = \frac{T}{n} = \frac{W\left(\frac{re}{p_f}\right)}{1 - W\left(\frac{re}{p_f}\right)} \tag{41}$$

We can write it in terms of premium $\lambda = \frac{p_f}{p_r}$

$$T_{opt} = \frac{W\left(\frac{e}{\lambda}\right)}{1 - W\left(\frac{e}{\lambda}\right)} = \frac{1}{\frac{1}{W\left(\frac{e}{\lambda}\right)} - 1} \tag{42}$$

Note that as $\lambda > 1$, $\frac{e}{\lambda} < e$, so $0 < W(\frac{e}{\lambda}) < 1$ and therefore $t_0 > 0$, and that $t_0(\lambda)$ is a monotonic decreasing function, with $\lim_{\lambda \to 1} t_0(\lambda) = +\infty$.

TODO: Prove that it's greater than break even time?

4.3.2 Approximating the optimal rebonding time

TODO: give some intuition why this is a good approximation. Is it because the APR is similar to the ARR for the considered time period? And apparently, maximizing the ARR yields the same result as the exact formula, as numerically verified by Dani.

As it turns out that the optimal rebonding time has no algebraic solution, we can also try to approximate it by finding the point the time that maximizes the APR.

We first define the *absolute premium* ρ as the difference between the market (or fair) price and the redemption price, i.e.: $\rho = p_f - p_r$.

Using the function $a(t)$ to express APR

$$a(t) = \frac{s(t) \cdot p_f - b}{b}\frac{365}{t} = \frac{\frac{b}{p_r}\frac{t}{t+\alpha}p_f - b}{b}\frac{365}{t}$$

$$a(t) = \left(\frac{p_f}{p_r}\frac{t}{t+\alpha} - 1\right)\frac{365}{t} \tag{43}$$

we derive and equal to zero:

$$a'(t) = \frac{p_f}{p_r}\frac{t+\alpha-t}{(t+\alpha)^2}\frac{365}{t} - \left(\frac{p_f}{p_r}\frac{t}{t+\alpha} - 1\right)\frac{365}{t^2}$$

(we assume $t > 0$)

9

$$a'(t) = 0 \iff \frac{p_f}{p_r}\frac{\alpha}{(t+\alpha)^2} = \left(\frac{p_f}{p_r}\frac{t}{t+\alpha} - 1\right)\frac{1}{t}$$

(we assume $p_r > 0$)

$$a'(t) = 0 \iff \alpha p_f t = p_f t(t+\alpha) - p_r(t+\alpha)^2$$

$$a'(t) = 0 \iff 0 = p_f t^2 - p_r(t^2 + 2\alpha t + \alpha^2)$$

$$a'(t) = 0 \iff (p_f - p_r)t^2 - 2\alpha p_r t - \alpha^2 p_r = 0$$

And solving for t (and getting the positive value):

$$t = \frac{2\alpha p_r + \sqrt{(2\alpha p_r)^2 + 4(p_f - p_r)\alpha^2 p_r}}{2(p_f - p_r)}$$

$$t = \frac{\alpha p_r + \sqrt{\alpha^2 p_r^2 + \alpha^2 p_f p_r - \alpha^2 p_r^2}}{p_f - p_r}$$

$$t = \frac{\alpha p_r + \sqrt{\alpha^2 p_f p_r}}{p_f - p_r}$$

$$t = \frac{\alpha p_r + \alpha\sqrt{p_f p_r}}{p_f - p_r}$$

$$t = \alpha\frac{p_r + \sqrt{p_f p_r}}{p_f - p_r} \tag{44}$$

In terms of the λ defined in 4:

$$t = \frac{\alpha\left(1 + \sqrt{\lambda}\right)}{\lambda - 1} \tag{45}$$

Comparing 45 and 33, we can immediately see that chicken in optimal time for APR is always strictly greater than break even time.

Simulations show a very close result to result from previous section for optimal rebonding time.

4.4 Effect of chicken ins and rebonding on the fair price

TODO: update once we have a formula for the fair price

4.5 Self-reinforcing price impact

It follows from equations 42 that a higher premium λ leads to a shorter rebonding period t (the same is true for 45, and for break even time in 33). As a consequence, we can expect bonders to chicken in earlier, resulting in a higher fraction of the bond b_p that transitions into the Permanent Bucket, which has a reinforcing effect on the premium and is favorable for the fair price of bTKN (see TODO).

Thus, if the market price of bTKN increases while everything else stays equal, the chances are higher that the price will continue to rise in the future thanks to earlier chicken ins. While this effect may be not always be immediate, a large bTKN price hike could effectively push bonders beyond their optimal rebonding times. Assuming perfectly rational behavior, this would result in immediate chicken ins (and rebonding).

TODO: check and update once we have a formula for the fair price

5 Protocol enhancements

5.1 Bootstrapping

Right after launch, the initial bTKN supply will be 0 as the token is only minted upon chicken in events. Similarly, the Acquired Bucket is empty until the first bond holder chickens in. As a result, the redemption price p_r (backing ratio) is undefined and cannot be used for calculating the accrued amount $s(t)$ during this bootstrapping period.

Meanwhile, the protocol already starts earning yield on the Pending Bucket before anybody chickens in. For fairness reasons, we want to ensure that this initial yield isn't simply earned by the bond holder who chickens in first.

To tackle this bootstrapping problem, we define the initial redemption price at the time of the first chicken in to be $p_r = 1$. The protocol thus needs to mint the same amount of bTKN as the amount of TKN acquired by the protocol up to that time. The accumulated yield of the Pending Bucket, which would normally belong to the Acquired Bucket, can be used to bootstrap a bTKN/TKN DEX pair, as the protocol can mint the bTKN to fill the other side of the AMM.

To make sure that the initial AMM price reflects the initial redemption price of 1, the system needs to deposit the same nominal amount of bTKN and TKN to the respective pool. So, it can use 50% of the accumulated yield (in TKN) to bootstrap the AMM, while moving the other half into the Acquired Bucket. The part of the yield used to bootstrap the AMM is technically also considered as being "acquired" and is thus reflected in q_a, such that $q_a = S$ holds upon the very first chicken in, ensuring $p_r = 1$.

5.2 Two-bucket version

A Permanent Bucket may not be suitable for all use cases and protocol tokens. For example, if TKN is a token that is minted through loans, keeping a portion of it inside the Permanent Bucket forever would essentially reduce the circulating supply, which could impact the ability of borrowers to repay their debts. On the other hand, such tokens have usually no fixed or capped supply, but can be minted as needed, meaning that technically there's an unbounded amount of TKN that could be bonded over time.

This makes bonding more sustainable in the long run even without a Permanent Bucket.

5.3 Auto-adjusting the steepness α of the accrual curve

TODO

5.4 Redemption fees

TODO

5.5 Payout tax as an incentive for the bTKN/TKN DEX pair

TODO