eBTC: Pre-Launch System Analysis

The Risk DAO

May 22, 2023

DISCLAIMER

The content is for informational purposes only, you should not construe any such information or other material as legal, tax, investment, financial, or other advice. Nothing contained on our report constitutes a solicitation, recommendation, endorsement, or offer by the Risk DAO or any third party service provider to buy or sell any securities or other financial instruments in this or in in any other jurisdiction in which such solicitation or offer would be unlawful under the securities laws of such jurisdiction.

All Content on this report is information of a general nature and does not address the circumstances of any particular individual or entity. Nothing in the report constitutes professional and/or financial advice, nor does any information on the report constitute a comprehensive or complete statement of the matters discussed or the law relating thereto. The Risk DAO is not a fiduciary by virtue of any person's use of or access to the report or content. You alone assume the sole responsibility of evaluating the merits and risks associated with the use of any information or other content on the report before making any decisions based on such information or other content.

Contents

1 Overview

The Badger DAO engages with the Risk DAO to review and analyze their eBTC system design. eBTC is an ERC20 that can be minted with (staked) ETH collateral and is designed with the intent of maintaining a peg to the price of BTC.

The goal of the analysis is to review liquidation mechanism, minimum collateral ratios, and the redemption mechanism.

1.1 Introduction

eBTC is a lending protocol that let users borrow eBTC, a BTC-pegged stable coin, against a staked ETH collateral. In return, the borrowers pay some of their staked ETH yield to the protocol. When a user collateral does not adhere to the minimum collateral to debt ratio (MCR), the user account is subject to *liquidation* where anyone can repay the user debt, and in return get the user collateral, with some discount on the BTC/staked ETH exchange rate.

Further, as another mechanism to maintain the peg, the protocol supports a *redemption* mechanism that let anyone redeem his or her eBTC in return to an equivalent amount of staked ETH (minus some varying fees).

While at its core the protocol resembles to the Liquity protocol [7], it has three different fundamental properties, which leads to different analysis of its parameters.

- The ETH/BTC pair is highly correlated in price, and thus fewer liquidations are expected.

- As fewer liquidations are expected, the eBTC will be launched without a stability pool.
- The protocol is governed by a DAO, and thus some of the parameters could be adjusted over time.
- The protocol has an effective interest rate (in the form of staked ETH yield sharing) and 0 one time minting fees.

1.2 Goals of the analysis.

Our review focuses in the following aspects:

1.2.1 Liquidation mechanics

We investigate the following topics:

- *Liquidation incentive.* Liquidators receive a discount on the market price when performing a liquidation. In some DeFi platforms, most notably Compound [8] and Aave [1] this incentive is fixed. While in others, e.g., Liquity [7] and Euler [6] the incentive depends on the current collateral ratio of the account. We demonstrate how the two options leads to different capital efficiency (MCR values) and bad debt accumulation. We also investigate an hybrid combination of the two approaches.
- *Absorbing bad debt.* In Liquity, failed liquidations are absorbed by distributing the position collateral and debt among the other borrowers. We investigate refined models where underwater accounts are still subject to liquidations, and only when the collateral is close to zero, then the remaining bad debt is distributed among the other borrowers. We compare the two approaches and provide simulation results.
- *Stability pool.* We study the need for a stability pool in the liquidation process.
- *Specific parameters.* We simulate how the system behave for specific set of parameters.

1.2.2 Redemption mechanism and interest rate

Keeping eBTC overcollateralized helps maintain the peg in the long run. However, liquidity crunches, and minting bursts could temporarily create an excess of supply, and move the price below 1.

Redemption helps bouncing the price back to 1, and we investigate the trade-offs of putting certain restrictions on redemption speed, and a mechanism that could improve the trade-off.

Interest rate is also an instrument to regulate excess and shortage in supply, and we plot the different considerations when deciding on the interest rate.

2 The Quantitative Simulation Model for Liquidations

Stable coin liquidity heavily relies on the properties of Curve Finance's stable swap formula. We ran our simulation model for liquidations of stable assets, and visualised how the curve inventory size, and the re-pegging velocity affects the ability to liquidate.

We consider a setup, where eBTC is traded vs WBTC on Curve [1], and WBTC is traded vs (staked) ETH. Further, the system always prices eBTC at 1 BTC (and we further assume that WBTC is also priced at 1 BTC).

During a liquidation process, eBTC is supplied in return to ETH collateral, and then the liquidator immediately sells the ETH back to eBTC. The ETH to eBTC sale is typically done by first selling ETH to WBTC, say on Uniswap, and then selling WBTC to eBTC on a Curve.

We assume that for a liquidation bonus of x%, the liquidator can execute the liquidation if and only if she can sell on Uniswap and Curve with an aggregated slippage of x% (i.e., the price of the last wei should not be lower by more than x% from the oracle price).

2.1 Simulation run

In a run, we plot:

- Price trajectory (in red)
- Liquidation sequence (blue)
- Open liquidations (yellow)
- The eBTC vs WBTC price in Curve (green)
- The eBTC balance in curve (purple)
- The total liquidity that is available for ETH to eBTC liquidations, for the given liquidation penalty (black)

In a liquidation, the liquidator has to buy eBTC, and therefore it immediately increases the price on Curve. However, when eBTC is traded above 1 BTC, it is unclear how long it will take to re-peg. In the absence of any physical constraints (e.g., DAI's PSM), we observed that it could historically take days and even weeks (e.g., LUSD).

We assume an exponential decay model, where the Curve imbalance is slowly decaying with a half life period of X days, i.e., after X days, it will be halved (assuming no further liquidations).

In the simulation depicted in Figure 1, the first few liquidations decreased the eBTC balance in Curve, but did not have a big impact on the price. Hence,

[1] The same analysis would hold also for Balancer stableswap formula

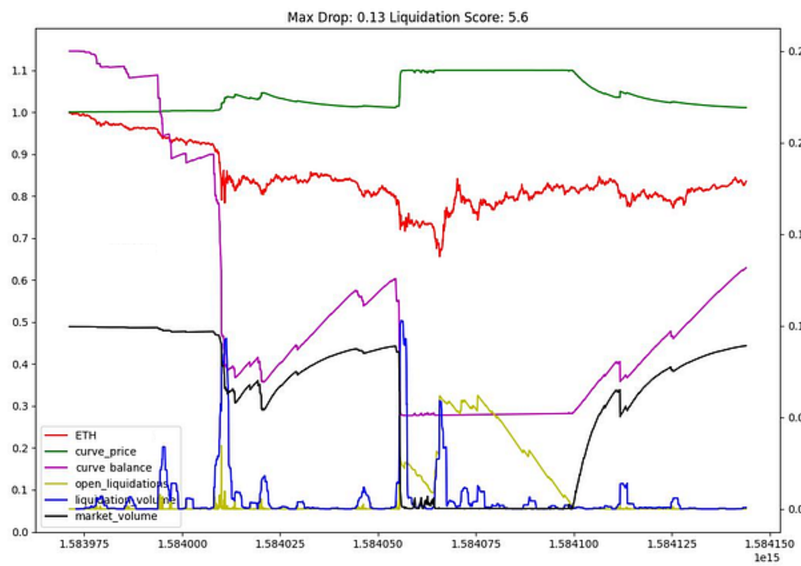


Figure 1: A single simulation run.

the available liquidity for liquidations was still dominated by the WBTC/ETH liquidity. However, additional liquidations occurred before the Curve balance retained to normal ratio. And the later liquidations choked the Curve pool, and effectively drained the available liquidity.

In Figure 2, we double the half life parameter, namely, the price recovery time is slower.

Now we see that the price does not re-peg until the end of the simulation, and as a result, some of the liquidations remain open until the end of the simulation.

Finally, in Figure 3 we ran the simulation with double the amount of Curve balance.

In this scenario, all the liquidations are handled in a timely manner, and the price of eBTC never exceeds 1.05.

2.2 Modeling liquidation events

We take liquidation events from real world data, namely, from Binance Futures exchange. We take the most popular asset that has a similar volatility as the asset we wish to simulate. In this work, we define similar as at least x0.2 and at most x5 times the volatility, and we defer the formal definition of volatility to the next subsection. We denote this sequence of liquidations as the *reference liquidations*, and the liquidation asset as the *reference asset*.

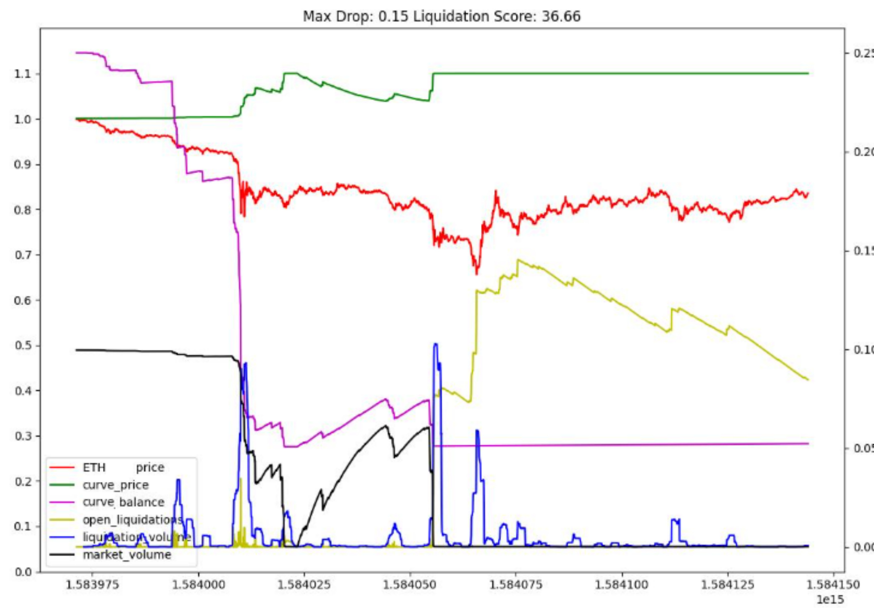


Figure 2: A simulation run with higher half life parameter.

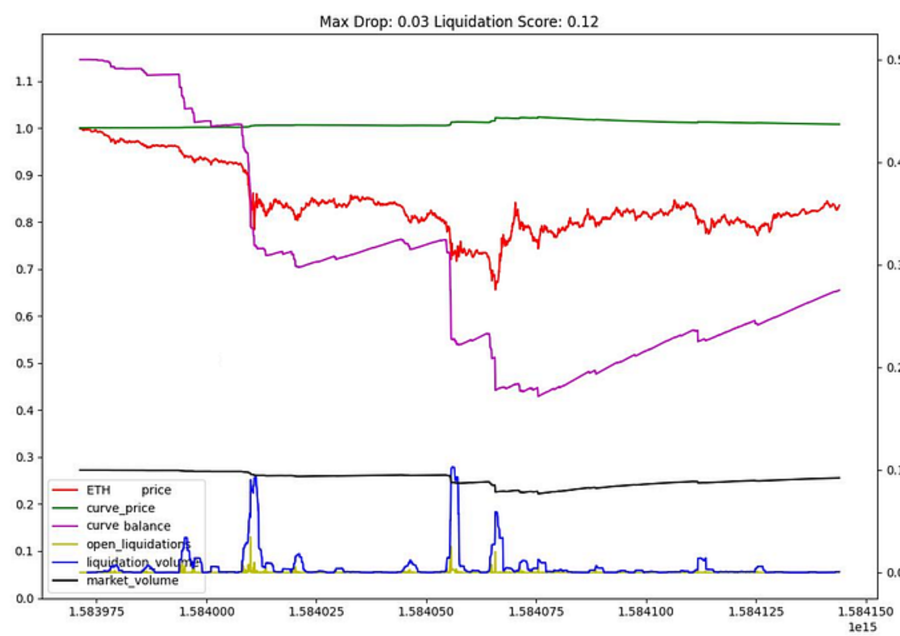


Figure 3: A simulation run with higher curve initial balance.

To choose the sequence of *simulated liquidation events* we ignore the volatility of the reference asset and simulated asset. Instead we fix the monthly liquidation volume, by setting a *simulation liquidation factor*, denoted by **slf**, and multiplying the liquidation volume of all reference liquidations by this factor.

We note that a simulated liquidation of volume v at time t means that there is a user position with collateral v at time t that is not sufficiently overcollateralized. The exact time in which the liquidation will take place is subject to the simulated market liquidity.

2.3 Modeling BTC/ETH price trajectory

We use price trajectories only to simulate the platform insolvency (and not to decide if liquidation will occur). Hence, we are mostly interested in how the price behaves between the time a user position is subject to liquidation (as described in the previous subsection) and the time the full liquidation is completed (as described in the next subsection). For this purpose we try to have a price simulation that is approximating short time frames. For a time duration of T minutes, we define the T *price average volatility* as the average T minutes price standard deviation, and denote it for an asset a by $\mathbf{vol(a)_T}$. Further, we define the *STD ratio* as the ratio $\frac{\mathbf{vol(sa)_T}}{\mathbf{vol(ra)_T}}$, where sa (respectively ra) is the simulated (resp. reference) asset. For example if the average volatility of SPELL is x2.8 higher than the one of ETH, then the corresponding STD ratio is 2.8. The average volatlity of ETH/BTC is half the one of ETH/USD, and therefore the ratio is 0.5.

Having the STD ratio in hand, we amplify, in every minute, the price change by that ratio. In our example, if between time t and $t+1$, the ETH price changed (either up or down) by 0.1%, then the simulated ETH/BTC price will change by 0.05%. Our numerical results show that if the STD ratio is under 5, then the simulated asset average volatility corresponds to its real world average volatility.

The intention of this process is not to simulate the long term price trajectory of the asset. But rather to sync the expected price changes along with the expected liquidations. In other words, we want that after every simulated liquidation the simulated price will behave similarly to the one of the reference asset after a reference liquidation. In Figure 4 we illustrate the short term simulated price for five different STD ratios. It shows that the price decreases are amplified when the ratio increases. The price increases are also amplified, however, as depicted in Figure 5, higher STD ratio skew the trajectory downwards. This stems from the fact that if one asset price movement is $(-5\%, +5\%)$, and a second asset price movement is $(-10\%, +10\%)$, then the second asset total price decrease is bigger. Overall the bias towards price decreases makes our simulation more conservative, and as depicted in Table 2.3 achieves our main goal, which is to create a simulated asset with the same average volatility.

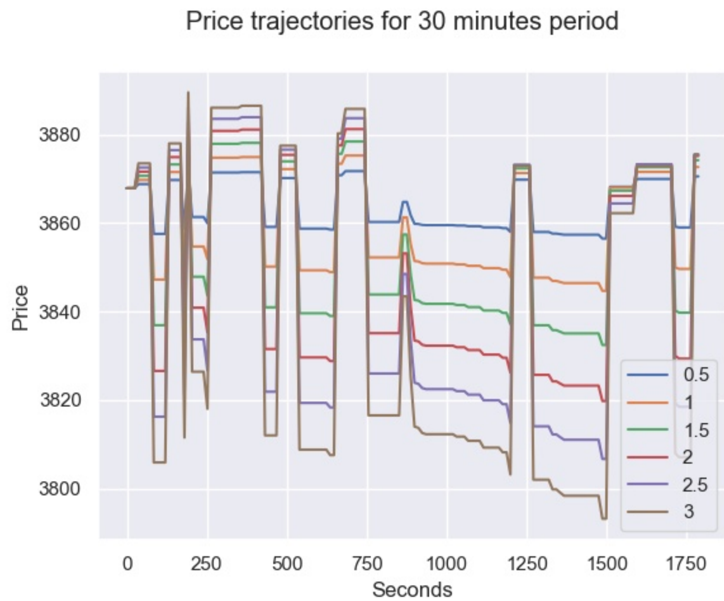


Figure 4: Simulated price trajectories for different STD ratio that range between 0.5 and 3.

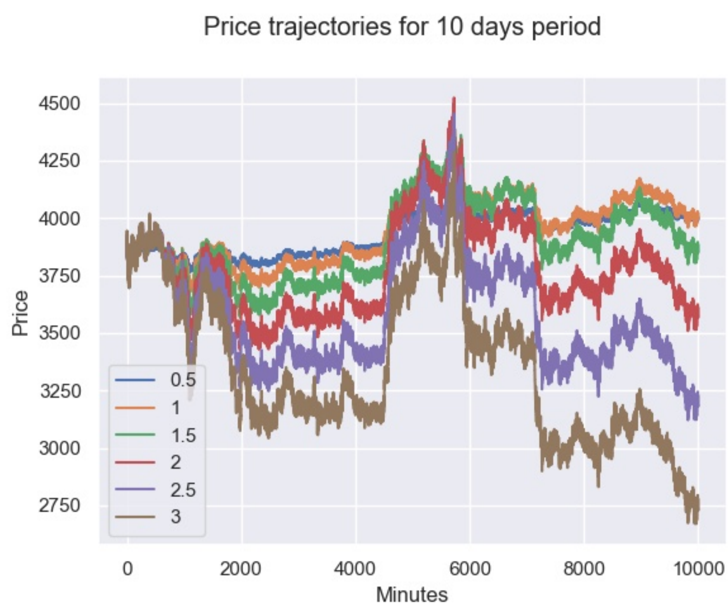


Figure 5: Simulated price trajectories for different STD ratio that range between 0.5 and 3.

STD ratio	30 minutes average volatility
0.5	0.499
1	1.000
1.5	1.500
2	2.0003
2.5	2.50007
3	3.001
4	4.004

2.4 Parameter selection

We fixed the STD ratio according to the last 6 month price volatility of ETH/BTC (and staked ETH assumed to have very similar volatility to the one of ETH). The liquidation trajectories are also fixed according to Binance futures historical data.

The input parameters for every simulation are:

- Curve initial balance for the WBTC/eBTC pair.
- Curve balance recovery half life.
- The available liquidity for (staked) ETH/WBTC.
- The liquidation stress factor.

As the eBTC system is not live yet, we ran it with different configuration inputs in order to reason about the research questions from Section 1.2.1.

3 Liquidation Simulation results

3.1 Liquidation bonus mechanism

The first question we investigate is a fixed liquidation bonus vs a bonus that depends on the current collateral to debt ratio of the account.

We denote the account collateral to debt ratio with ICR (individual collateral ratio), and the liquidation bonus by β, where $\beta = 0.1$ means that for a liquidation of \$$x$ debt, the liquidator would get \$$1.1x$ in collateral. We denote the *minimum collateral ratio* by MCR, and whenever the ICR < MCR, then the account is subject to a liquidation.

It is an easy observation that for a fixed liquidation bonus β, if ICR $< (1+\beta)$, then the liquidation would end up with bad debt (i.e., with zero collateral, and non zero debt). And therefore the MCR must be greater than $(1 + \beta)$.

Therefore, if we aim for an MCR of 1.1, a fixed liquidation bonus will dictates a bonus smaller than 0.1, and if we aim for a bonus of 10%, then an MCR that is higher than 1.1 is needed.

We therefore investigate and compare the simulation results of the following three options:

1. MCR = 1.1; $\beta = 0.05$.
2. MCR = 1.2; $\beta = 0.1$.
3. MCR = 1.1; β = min(ICR, 1.1) - 1.

The first two options represent fixed liquidation bonuses, and the other one is a varying bonus. We note that the last option might choke when ICR < 1, and we investigate a ways around it in the next section.

For the purpose of the analysis we measure two different metrics:

- *Max drop.* The maximum price decrease from the time an account was subject to liquidation until the time the liquidation was fully executed. The decrease includes the liquidation bonus, and if max drop $>$ MCR - 1 it means that bad debt was accrued.
- *Liquidation score.* This is the integral of open liquidation amounts. Meaning, that if a liquidation of amount 100 was open for 8 time units, then it will contribute 800 to the score. For simplicity we divide the integral by the total duration of the simulation.

The goal is to select parameters that minimize both of these metrics. While minimizing bad debt (max drop) is the ultimate goal, the liquidation score also reflects risks that did not materialize, and could be materialized, if, e.g., BTC/ETH volatility spikes. Hence, it is desirable to try and minimized it as well.

Partial simulation results are depicted in Figure 6, and the full results for additional parameters are available here [2] and they are similar to the ones that are depicted in the heatmaps.

As expected, the lowest score is always when the liquidation bonus is higher (0.1), as in such a scenario, the liquidations will be executed quicker. Thanks to the low volatility of the ETH/BTC pair, we get the lowest max drop for β = min(ICR, 1.1) - 1. Finally, we see that a bonus of 0.05 always under performs in both max drop and score. Hence, the surface of desirable configurations is either to have a fixed β with higher MCR (option 2) or varying β with lower MCR (option 3). While option 2 gets better score, the difference is at most 6%. Giving the system is in a pre-launch phase, and there are still many unknowns about the actual liquidity and price behavior, we believe that 6% difference is not meaningful, and therefore having a liquidation bonus that depends on the ICR is the most viable option [2].

3.2 MCR value

As the system is at a pre-launch state and the exact market conditions are currently unknown, we ran the simulation for 2,700 different configurations (see

[2] under the assumption there are additional mechanisms to absorb bad debt when ICR $<$ 1.

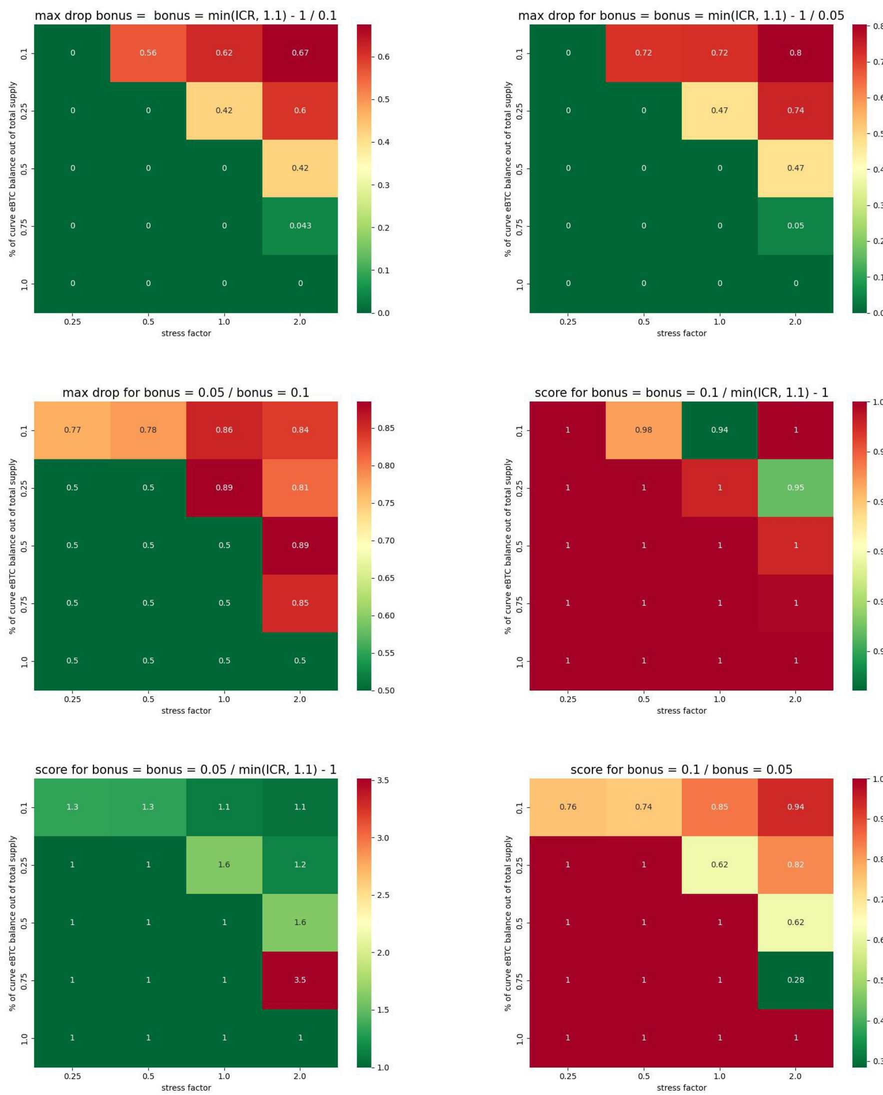


Figure 6: Max drop and simulation score for half life recovery time of 1 day and ETH/WBTC liquidity of 0.1. The y axis is the curve initial balance, and the x axis is the stress factor. A value bigger than one means the results of the left hand bonus are better.

Section 2.4). In order to present the results in a human readable way, we present the minimal conditions to support the desirable MCR range.

Formally, recall that a configuration input is a 4 dimensional vector. Higher liquidity/curve balance is always preferable over lower ones. Similarly. shorter price recovery half life, and lower stress factor will also give better results, and hence are more preferable. For two configurations c_1 and c_2, we say that $c_1 < c_2$ if each dimension in c_1 is not preferable to its corresponding dimension in c_2. For example, c_1 has worse liquidity, longer half life, and equal to c_2 in the other two parameters. For a given MCR range, we say that a configuration c is minimal, if for every other configuration d in the range: $d \not< c$.

Intuitively, it means that in order to improve the simulation results, we must improve one of our input parameters, meaning to relax our assumptions about the system state.

The full simulation results are browsable via [3].

Table 1 presents the minimal conditions to support MCR of 1.1. The stress factor ranges between 0.1 and 2. Where a factor of 0.25 simulates total monthly liquidation volume of 25% of the entire supply. Hence, the relevant range under normal conditions is 0.1-0.5. Therefore, the simulations show that an initial curve eBTC balance of 25-50% of the total eBTC supply would be able to support a MCR of 1.1. Interestingly enough, we see that increasing the ETH/WBTC liquidity beyond 0.1% of total eBTC supply would not improve the results. At the time of writing, (staked) ETH/WBTC liquidity for 10% slippage is around $50M. Hence, this liquidity will suffice even for an eBTC supply of $500M.

Liquidations are the first line of defence against bad debt events. When liquidations fail, the system redistribute the bad debt to all other borrowers. Hence, when the *total collateral ratio* (TCR) of the users is low, the ability to absorb the bad debt is lower. Hence, eBTC sets an higher MCR when the TCR is low and the system is in recovery mode. The recovery mode MCR is denoted by CCR (critical collateral ratio). In Liquity system, the CCR is 1.5. Given the ETH/BTC pair is more correlated than the USD/ETH pair, we examine the possibility to set a CCR of 1.25 instead of 1.5. For this purpose, we examine the minimal configurations to support an MCR of 1.25 (Table 2), and to support MCR 1.5 (Table 3). We see that in general, an MCR of 1.5 is needed either when the curve liquidity is below 0.25, or when the stress factor is above 1.25. Hence, we recommend to start with an CCR of 1.25.

3.3 Absorbing bad debt

Having a liquidation bonus of min(ICR,1.1) - 1 dictates that liquidations would not be executed when the ICR < 1. Therefore, whenever there is an account with ICR < 1, there is a bad debt event, which can only be wiped out if the position is absorbed by the other borrowers (like in Liquity) or if the price of eBTC increase.

Redistributing the position means that the insolvent account collateral and debt are distributed, proportionally, among all the other borrowers (propor-

Curve initial balance	Half life time	ETH/WBTC liquidity	Stress factor	max drop
0.1	5	0.1	0.1	0
0.1	1	0.1	0.25	0
0.25	5	0.1	0.25	0
0.25	2	0.1	0.5	0
0.25	0.5	0.1	0.75	0
0.5	5	0.1	0.75	0
0.5	2	0.1	1	0
0.75	5	0.1	1	0
0.5	1	0.1	1.25	0.03
0.75	4	0.1	1.25	0.06
0.75	2	0.1	1.5	0.09
1.0	5	0.1	1.5	0.09
0.5	0.5	0.1	1.75	0.09
0.1	0.001	0.1	2	0.09
0.75	1	0.1	2	0.09
1.0	2	0.1	2	0.09

Table 1: Minimal configurations to support MCR 1.1.

Curve initial balance	Half life time	ETH/WBTC liquidity	Stress factor	max drop
0.1	5	0.1	0.1	0
0.1	1	0.1	0.25	0
0.25	5	0.1	0.25	0
0.25	2	0.1	0.5	0
0.25	0.5	0.1	0.75	0
0.5	5	0.1	0.75	0
0.5	2	0.1	1	0
0.75	5	0.1	1	0
0.1	0.001	0.1	1.25	0.197421
0.5	1	0.1	1.25	0.197421
0.75	4	0.1	1.25	0.237667
1	5	0.1	1.25	0.197421
0.75	2	0.2	1.5	0
1	5	0.2	1.5	0
0.5	0.5	0.2	1.75	0
0.1	0.001	0.2	2	0.084494
0.75	1	0.2	2	0.237667
1	2	0.2	2	0

Table 2: Minimal configurations to support MCR 1.25.

Curve initial balance	Half life time	ETH/WBTC liquidity	Stress factor	max drop
0.1	5	0.1	0.1	0
0.1	1	0.1	0.25	0
0.25	5	0.1	0.25	0
0.1	0.5	0.1	0.5	0.454065
0.25	2	0.1	0.5	0
0.25	1	0.1	0.75	0.401548
0.5	5	0.1	0.75	0
0.5	2	0.1	1	0
0.75	5	0.1	1	0
0.25	0.5	0.1	1.25	0.454065
0.75	4	0.1	1.25	0.237667
0.5	1	0.1	1.5	0.454065
0.75	2	0.1	1.5	0.324663
1	5	0.1	1.5	0.324663
0.1	0.001	0.1	2	0.328068
0.5	0.5	0.1	2	0.454065
0.75	1	0.1	2	0.328068

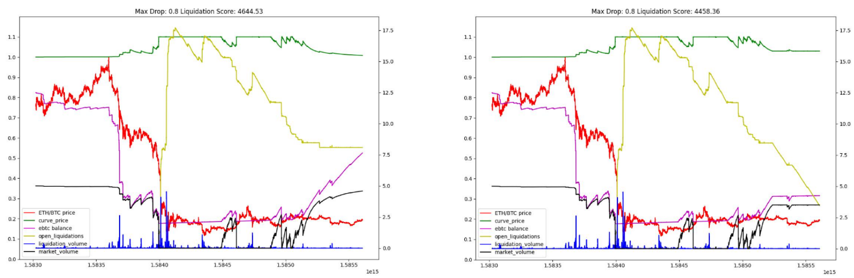
Table 3: Minimal configurations to support MCR 1.5.

tionally to their debt). As the ICR of the insolvent account is less than 1, this process can be quite painful to the other borrowers in the event that the account is big. Moreover, this will decrease the individual ICR of each of the other borrowers.

We therefore suggest that the liquidation bonus should always be positive (even when ICR < 1). This can be done by setting $\beta = \max(1.03, \min(\text{ICR},1.1))$. We note that having such minimum bonus had very little affect on the simulation results, with the current ETH/BTC volatility (see full results here [4]) however, when we artificially amplified the volatility, the affect was more visible. In Figure 7 we plot two simulation runs. One with a minimum bonus of 3%, and the other without any minimum incentive. While the max drop result is similar in both, the open liquidations (yellow line) remain open when there is no minimum bonus, which increase the overall risk of the system.

Further, we provide a short proof if the TCR $> (1+\beta)$, then a (partial) liquidation improves the TCR, even when the account is insolvent. Let c be the collateral size of the liquidated account, d the total debt of the account. C the total collateral in the system, and D the total debt in the system. Then during a liquidation, d' of the debt is paid, and in return the liquidator receives $(1+\beta)d'$ collateral. Hence, before the liquidation, the TCR is $\frac{C}{D}$, and after the liquidation it is $\frac{C-(1+\beta)d'}{D-d'} = \frac{C}{D} \cdot [\frac{DC-(1+\beta)d'D}{DC-dC}]$. Since the TCR $= \frac{C}{D} > (1+\beta)$ we get that $C > (1+\beta)D$, and the proof is complete.

Hence, we recommend to maintain non zero liquidation bonus, and to dis-



(a) (b)

Figure 7: On the left hand side, a simulation run with $\beta = \min(\text{ICR},1.1)$. On the right hand side, a simulation run with $\beta = \max(1.03, \min(\text{ICR},1.1))$.

tribute the insolvent position among the other borrowers only when the collateral is zero, or too small for the liquidation bonus to cover the gas costs. We note that since the system give a fixed gas stipend for a full liquidation, the bonus + stipend should always make a liquidation profitable, and thus the concern that a dust amount of collateral will remain, and will prevent the debt from being absorbed is mitigated.

3.4 Stability pool

The eBTC design does not include a native stability pool. In Liquity's original design, users deposit eBTC into a pool, and whenever a liquidation is needed the eBTC is taken from the pool, and the depositors receives the liquidated collateral instead. Naturally, every eBTC in the stability pool decreases the likelihood of a failed liquidation, as it remove the dependency on the market's ETH/WBTC and WBTC/eBTC liquidity. However, the current ETH/WBTC liquidity is pretty decent. And as we observe in Table 1, the current liquidity could support up to \$0.5B eBTC supply. Hence, the WBTC/eBTC liquidity becomes the bottleneck, towards successful liquidations. Figure 8 shows that 80% of the eBTC in curve can be used for liquidations, before the price hits 1.1. Hence, the efficiency of an eBTC in the stability pool is only 25% higher of the one in curve. Further, Liquity's history suggest that a big stability pool only increases the upwards pressure for eBTC price, which in turn delays the price recovery after a liquidation.

On the negative side, it should be noted that any eBTC in curve must be paired with an additional WBTC deposit. And hence the total needed capital is doubled. However the curve liquidity also facilitates the day to day trading activity, and thus it makes sense to launch the system without a stability pool.

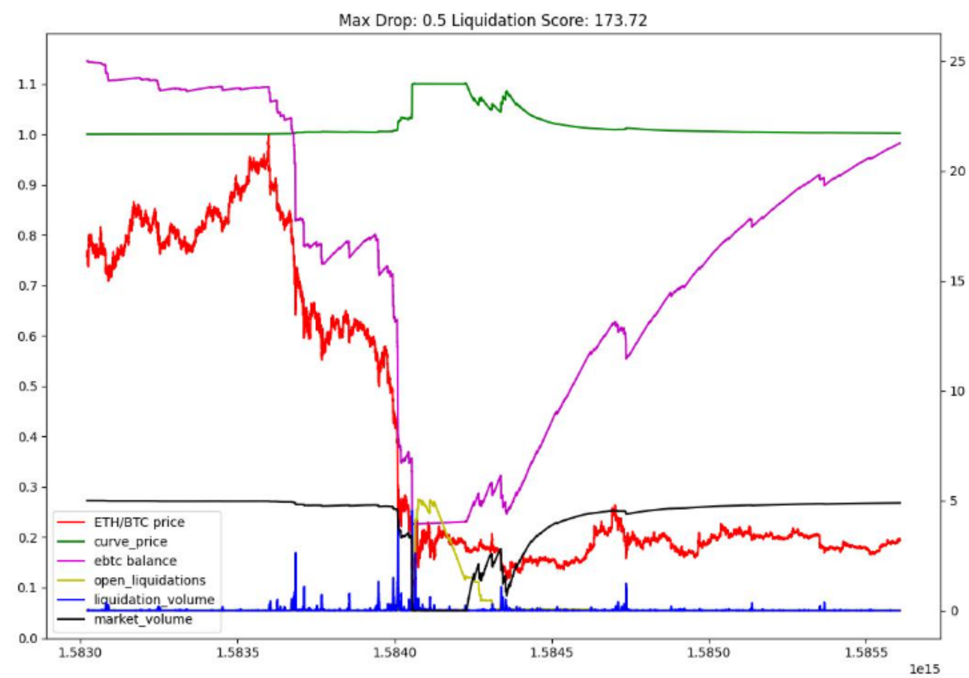


Figure 8: A simulation with an initial curve balance of 25 units. The eBTC/WBTC price hits 1.1 (green line) when the balance drops to 5 units (purple line).

4 Modeling Redemption

Redemptions play a key role in reassuring the community that the price of eBTC will not drop much below 1 BTC (as long as the system does not have bad debt). However, redemption comes with a cost. Every redeemed eBTC comes at the expense of a lower total supply, which damage both the adoption of the protocol and accumulated fees.

The Liquity model can cap the redemption speed, and we investigate whether this speed should be bounded or not. Intuitively, an unbounded speed guarantees hard floor to the price of eBTC, but will also increase the amount of total redemptions, and thus, decrease the total supply the most. Bounded speed weakens the floor, however, it gives some time for the price to organically recover, and thus can potentially decrease the redemption total volume.

Formally, we build a simulation model where the mint demand and buy demand are randomly sampled in such way that on average mint = buy. We note that if for the long run the mint demand is higher than the buy demand, then reducing the supply, ideally via interest rate, but potentially also via redemption, is a bug and not a feature. For simplicity, we assume there is a fixed daily redemption limit (could also be infinity), and whenever the price is below 0.98 BTC, arbitragers will try to redeem eBTC, up to the daily limit. We do the simulations for different initial curve balances and different daily limits (as % of the total curve balance).

Naturally, redemption caps would result in fewer redemptions, however when traded below 1, minting new eBTC and selling it on the open market is not attractive, as the minter has to sell with a discount. Lower mint incentive also damages the total supply, and we simulate the outcome with different penalty functions.

Finally, we also introduce a theoretical sink mechanism, in the form of an option box, to temporarily absorb eBTC excess supply. This sink allow users to stake eBTC with a future promise to redeem then for 0.98 BTC (in ETH) at a later date.

We run Monte Carlo simulations for different parameters and distributions. Overall it shows that unlimited redemption speed is the least desirable option. It also shows that the sink mechanism can give rise to a significant improvement, however the numerical results heavily depends on our simulation model, and the exact user behavior is also less certain. Hence, we recommend to launch without it, and to consider it as future improvement to the protocol, if it suffers from long deppeging periods.

4.1 Simulation model and results

4.1.1 Distribution Assumptions

For starters, we will assume a random flow of minters (assumed to be sellers) and buyers of stablecoin within a probability space (Ω, F, P), where $x_0, x_1, x_2, \ldots, x_n | Ft, t > 0$ acts as a sequence of i.i.d random variables that are normally distributed (formally denoted as $N(\mu, \sigma^2)$ within infinitesimal timestamps, where x_n = value of x at timestamp n).

Our SDE (stochastic differential equation) is the following:

$$dX_t = \mu(X_t)dt + \sigma(X_t)dW_t$$

Where Wt is a $Ft, t > 0$ measurable n-dimensional Wiener Process, essentially creating a perpetual element of "randomness" as it relates to our parameters chosen (more importantly, the diffusion term)

Or, in integral form as

$$X_t = X_0 + \int_0^t \mu(s, X_s)ds + \int_0^t \sigma(s, X_s)dW_s$$

,

Where $0 < s < t$, and the time increment (W_t - W_s) is normally distributed with $\mu = 0$ and σ^2 = (t - s), formally denoted as (W_t - W_s) $\sim$ N(0, t - s)

We assume our μ (mean) = 0 (i.e. volume flow assumption is neutral, aka "martingale") whilst σ^2 (variance) is linked to our standard deviation (σ) assumptions derived below, of which are linked to monthly volume operations. As such, we must prove that the sum of normally distributed random variables $(x_1, x_2, x_3, \ldots, x_n)$ is also normally distributed (i.e. each x_i is a normally distributed random variable with mean μ_i and variance σ_i^2, using the characteristic function given by

$$\varphi_{X_t} = e^{it \cdot \mu - (t^2\sigma^2)/2}$$

Defining S as the sum of the normally distributed random variables:

$$S = x_1 + x_2 + x_3 + \ldots + x_n$$

Of which the characteristic function of S acts as the product of the characteristic functions of each xi:

$$\varphi_{S_t} = \varphi_{x_1(t)} \cdot \varphi_{x_2(t)} \cdot \varphi_{x_3(t)} \cdot \ldots \cdot \varphi_{x_n(t)}$$

Substituting the norm. dist. characteristic function into the above equation, we get:

$$\varphi S_t = e^{it\mu_1 - (t^2\sigma_1^2)/2} \cdot e^{it\mu_2 - (t^2\sigma_2^2)/2} \cdot \ldots \cdot e^{it\mu_n - (t^2\sigma_n^2)/2}$$

Taking the logarithm of both sides:

$$\ln(\varphi S_t) = \ln(e^{it\mu_1 - (t^2\sigma_1^2)/2)}) + \ln(e^{it\mu_2 - (t^2\sigma_2^2)/2)}) + \ldots + \ln(e^{it\mu_n - (t^2\sigma_n^2)/2)})$$

Simplifying the above equation (canceling out the log and exp terms and abstracting the (μ_i, σ_i^2) terms into general summation with respect to time), we get:

$$ln(\varphi S_t) = it(\Sigma_{i=1}^{n}\mu_i) - (t^2/2)(\Sigma_{i=1}^{n}\sigma_i^2)$$

Naturally, the above equation is the characteristic function of a normally distributed random variable with mean $(\Sigma_{i=1}^{n}\mu_i)$ and variance $(\Sigma_{i=1}^{n}\sigma_i^2)$, thus our sum of normally distributed random variables $(x_1, x_2, x_3, \ldots, x_n)$ is also normally distributed, of which from now on will be defined as $(\tilde{\mu}, \tilde{\sigma})$

As such, we now abstract the timestamps into respective monthly operations over an annual timeframe, where $n = 24$ daily timestamps x 30 days in a month $= 720$ timestamps, formally

$$x_1 + x_2 + x_3 + \ldots + x_{720} \subseteq x_1 + x_2 + x_3 + \ldots + x_{8760}$$

s.t. our net volume over the year-long distribution equates to

$$\mathbb{E}[280_{tot} \cdot eBTCcurveLPsize_0]$$

or

$$\mathbb{E}[140_{mint} \cdot eBTCcurveLPsize_0] + \mathbb{E}[140_{buy} \cdot eBTCcurveLPsize_0]$$

Given the sum of normally distributed random variables is normally distributed itself, and we know by the empirical rule in statistics that approximately 68% of the data will fall within one standard deviation of the mean (formally,

$$P(\tilde{\mu} - \tilde{\sigma} \leq X \leq \tilde{\mu} + \tilde{\sigma}) = \varphi(1) - \varphi(-1) = 0.8413 - 0.1587 = 0.6826)$$

, we set the probability of $|buy - mint| > 5 \cdot eBTCcurveLPsize_0$ to fall outside one standard deviation (or occurring in 31.74% of the months). Solving for our generalized $\tilde{\sigma}$ (or $\sqrt{\sigma^2}$) naturally becomes

$$\sigma = 5 \cdot eBTCcurveLPsize_0$$

4.1.2 Mean Reversion

To simulate the future redemption mechanisms in varying states, we now introduce a very simple "mean-reverting" process to gauge relative peg movements, such that $\tilde{\mu}$ (mint - sell differential) of a given month is *always* 0.

First, we let X be the sum of these random variables, i.e.,

$$X = x_1 + x_2 + x_3 + \ldots + x_n$$

.

Of which we know the expected mint - sell value = 0, thus by linearity of expectation:

$$\mathbb{E}[X] = \mathbb{E}[x_1] + \mathbb{E}[x_2] + \mathbb{E}[x_3] + \ldots + \mathbb{E}[x_n] = 0 + 0 + 0 + \ldots + 0 = 0$$

Plugging in $\tilde{\mu}$ as $= X/n$, the expected value of $\tilde{\mu}$ is:

$$\mathbb{E}[\tilde{\mu}] = \mathbb{E}[X/n] = \mathbb{E}[X]/n = 0/n = 0$$

Now, we can proportionally alter the values of $x_1, x_2, x_3, \ldots, x_{((1/12)n)}$ subset (such that the *realized* mean value of the monthly subset is also zero), through defining a new set of random variables $y_1, y_2, y_3, \ldots, y_n$, where

$$y_i = x_i - \tilde{\mu}$$

for all i.

Then, we have:

$$\mathbb{E}[y_i] = \mathbb{E}[x_i - \tilde{\mu}] = \mathbb{E}[x_i] - \mathbb{E}[\tilde{\mu}] = 0 - 0 = 0$$

Thus, the expected value of each of these new random variables is also zero, solidifying $\tilde{\mu}$ as 0 (through altering the relative i.i.d norm. dist values).

4.1.3 Curve Finance

You may be wondering - why don't we simply construct a probability space discreetly relating to the peg movements around 1? The answer lies within the Curve Finance "leveraged" stableswap bonding curve, heavily popularized by stablecoins/liquid staking derivatives looking to maximize liquidity in uniform automated-market-maker (AMM) fashion, such that the underlying AMM pool price at a given time t is directly linked to the pool's "amplification constant's" (denoted as A) relationship with the % balances of pool reserves:

In short, Curve utilizes an iterative combination of both a generalized constant sum ($\Sigma x_i = D$) and constant product ($\Pi x_i = (\frac{D}{n})^n$ market makers, respectively weighed (on a pool by pool basis, as it relates to the underlying asset(s) dependency to move away from 1, derived through governance) through the utilization of an Amplification constant, i.e. the level of liquidity concentration around the $1 price and its underlying effect on slippage/price impact for a given pool. Formally,

$$\chi D^{n-1} \cdot \Sigma x_i \cdot \Pi x_i = \chi D^n + (\frac{D}{n})^n$$

where

$$\chi = \frac{A \cdot \Pi x_i}{(\frac{D}{n})^n}$$

Plugging in χ and simplifying

$$An^n \cdot \Sigma x_i + D = ADn^n + (\frac{D^{n+1}}{n^n \cdot \Pi x_i})$$

[5]

Defining S as D/2, and solving for the amount of y tkns in the pool assuming inherently fixed D as a function of the amount of x tkns in the pool:

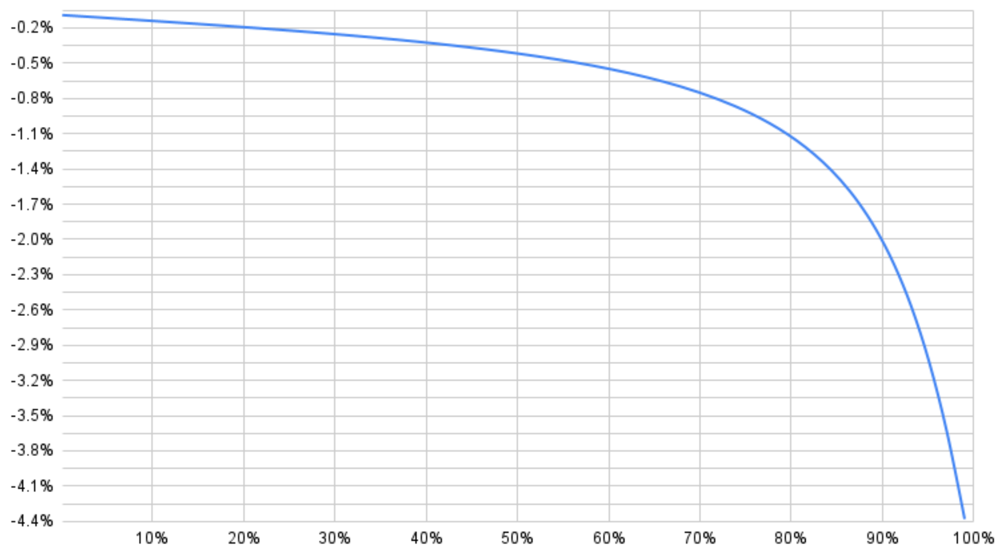


Figure 9: Swap slippage + fees as it relates to x in/initial x balance (assuming pool balance 50-50 @ t_0

$$ycurve(x) = \frac{-x}{2} - \frac{s}{4A} + s + \frac{\sqrt{(2Ax^2 + sx - 4Asx)^2 + 8Axs^3}}{4Ax}, x > 0$$

Solving for the derivative to find the slope :

$$y'curve(x) = \frac{(2Ax^3 + x^2s - 4Ax^2s - 2s^3)}{2x \cdot \sqrt{8Axs^3 + (2Ax^2 + xs - 4Axs)^2}} - \frac{1}{2}$$

Making y'curve(x) positive to find the underlying $pAMM$:

$$y'curve(x) \cdot -1 = pAMM$$

As you can see, pool state at t+1 can look very different from the net execution (or "average") realized by the swapper at a given time t. More importantly however, not only does the net flow of buyers and sellers within the probability space allow us to paint a more accurate representation of how the underlying price pAMM *reacts* under "normally distributed" liquid circumstances in either direction (or swaps), net directional volume required to affect pAMM when pool is unbalanced (or mean reversion) is greatly mitigated given non-uniform "leveraged" invariant properties (relating to the A factor)

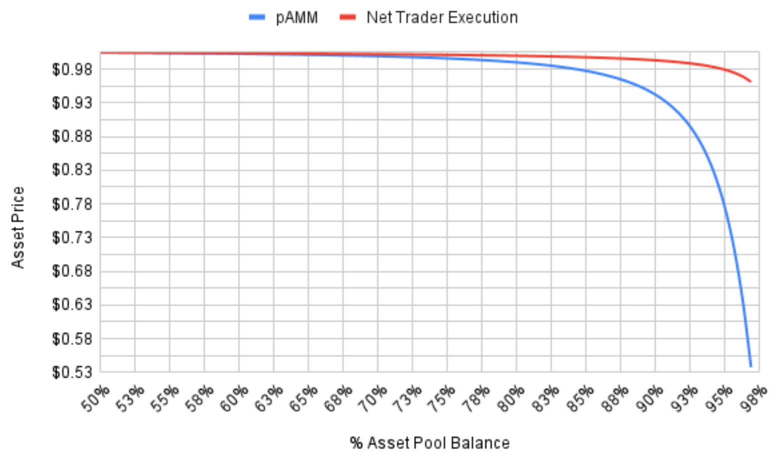


Figure 10: pAMM vs net trader execution as a function of % asset pool balance, assuming pool A factor = 100

4.1.4 Secondary Mean Reversion Term - Price Power Factor

While our integration of the Curve Finance stableswap invariant provided a more realistic outlook as it relates to peg fluctuations (as well as our mean reversion assumptions) - obviously, one cannot simply assume buy $\leftrightarrow$ sell (minter $\leftrightarrow$ buyer) pressure to be constant as our stablecoin/derivative fluctuates around the 1 mark; After all, if we fundamentally assume the price of the stablecoin/derivative to be 1, if pAMM > 1, buyers will naturally realize a haircut (and vice versa with minters if $pAMM < 1$). This ($pAMM \leftrightarrow pCDP$) obviously wildly depends on the underlying CDP mechanisms at hand (as mentioned earlier), though for simplicity's sake (as it relates to the peg mechanisms we will be modeling) we simply introduce a "Price Power Factor" of which we will denote as

$$\lambda \in \mathbb{N}\{0, 1, 2, 3, 4, 5\}$$

:

If $pAMM_t$,

$$< 1 = x_t \cdot pAMMt^{\lambda}$$

$$> 1 = \frac{x_t}{pAMMt^{\lambda}}$$

= adj. volume within said timestamp, or

$$\tilde{X}_t \in \mathbb{R}^+$$

Given redemptions act as a function of supply decay, to find the "Total Penalty" of the entire sequence, we add:

$$TotalPenalty = \Sigma_{i=1}^{T}(X_i - \tilde{X}_i) + total\ redemptions$$

(Note that "total redemptions" refers to respective redemption occurrence through the utilization of said redemption mechanisms).

$Totalpenalty/totalminting = netrealizedskew$ over the entire probability space.

4.1.5 Redemption fees (as a function of tot max Oracle deviation)

Given the utilization of Chainlink oracles to determine the underlying price of the stablecoin with respect to the collateral asset (and to avoid unfair redemptions due to oracle inaccuracies), for our models we assumed a total max deviation ϵ of 2% - thus we set the base hard redemption fee (as well as the "strike" for soft redemptions) at 98 cents. In reality however we will assume $\epsilon = 1.5\%$ (that is, unless Chainlink tightens the spread(s)) to account for the sum((BTC $\leftrightarrow$ ETH) + (ETH $\leftrightarrow$ staked ETH)) deviation thresholds).

4.1.6 Modeling Traditional Hard Redemptions (Redemption Process)

Now that we have developed our probability space and its various surrounding properties, we can finally start applying it to our redemption processes. For LUSD's hard redemption mechanism, we remember that an arbitrageur's redemption fee % incurred is linearly related to the % of LUSD supply redeemed:

$$b_t := b_{t-1} + \alpha \cdot \frac{m}{n}$$

[7]

where b_t is the base rate at time t, m the amount of redeemed LUSD, n the current supply of LUSD, and α is a constant set to .5

On the flip side, the redemption base rate exponential decay:

$$b_t := b_{t-1} \cdot \Delta_t^{\delta} t$$

where δ is the hourly decay factor s.t. half life = 12 hrs (thus .9438) whilst Δ_t is time(hrs) $\in \mathbb{R}^+$ since last redemption, until a $min(b_t)$ of .5%

As such, we simulate a similar property to our $\tilde{X}_t$ flow of buyers through the Curve pool, of which we simply set varying daily upper redemption bounds (limits) as a function of the total Curve eBTC liquidity (of which we estimate as some 50% of the eBTC supply at a given time t), or naturally until it is no longer economical to do so (in the form of slippage + swap/redemption fees incurred). Here, we define $\tilde{X}_t \in \mathbb{R}^+ \sim (0, \sigma^2)$ as y_{in}, or a redeemer buying eBTC (x asset) within a given timestamp.

As defined earlier,

$$xcurve(y) - xcurve(y + y_{in}) \cdot fees = x(eBTC)out$$

Power Factor VS daily redemption / curve initial balance	1%	10%	inf
0	0.17	0.35	0.4
1	0.31	0.42	0.46
2	0.4	0.48	0.51
3	0.49	0.54	0.57
4	0.56	0.6	0.62
5	0.63	0.65	0.67

Figure 11: Hard Redemption at 98 cents, no mean reversion, low volatility. Note that low volatility $= 3 \cdot \sigma$

for a given redeemer

$$\frac{x_{out}}{\tilde{X}_t} = redeemer\ realized\ execution/price$$

The pAMM however > redeemer realized execution/price, as:

$$pAMM = y'curve(x - x_{out}) \cdot -1$$

i.e. price with respect to the additional x out is a function of the highest realized avg redeemer value (as with traditional CPAMMs) for all "upper bound" values

The three columns denote the median % of $eBTCcurveLPsize_0$ redeemed over the course of the probability space (1 year), as it relates to daily redemption limitations (1%, 10%, and inf respectively). Our second parameter (underneath "Row Labels") denotes our λ "penalty" skew, creating a surface-like structure.

For our soft redemption mechanism, as mentioned earlier, we formally assume the volume assumptions (in any case) to be:

If $pAMM_t$

$< K_t$, *eBTC deposited/locked for* τ *days until maturity* T

$$> K_t, eBTC\ not\ deposited\ in\ options\ box$$

$$< K_T(maturity), all\ unlocked\ eBTC\ is\ redeemed\ at\ strike\ K$$

$$< K_T(maturity), all\ unlocked\ eBTC\ is\ dumped\ on\ Curve\ until\ pAMM{=}K,$$

$$then\ redeemed\ at\ strike\ K\ if\ applicable$$

With soft redemptions, our initial characterization based on daily % redemption limitations can no longer hold, thus we use a different parameter indicating values assuming various redemption lockup times/"maturity dates" in the form

Power Factor VS daily redemption / curve initial balance	1	10	inf
0	0.83	3	4.51
1	3.02	3.6	4.75
2	3.5	4	5
3	3.91	4.3	5.17
4	4.25	4.6	5.38
5	4.57	4.9	5.6

Figure 12: Hard Redemption at 98 cents, mean reversion, high volatility. Note that high volatility $= 10 \cdot \sigma$

Min Staking Time VS Power Factor	24	168	720
0	0.24	0.08	0
1	0.33	0.2	0.14
2	0.4	0.3	0.26
3	0.47	0.4	0.36
4	0.54	0.5	0.45
5	0.61	0.57	0.53

Figure 13: Soft Redemption at 98 cents, no mean reversion, low volatility.

Min Staking Time VS Power Factor	24	168	720
0	2.65	0.87	0.98
1	2.95	1.36	1.45
2	3.2	1.78	1.91
3	3.6	2.22	2.3
4	3.9	2.61	2.75
5	4.2	2.97	3.16

Figure 14: Soft Redemption at 98 cents, mean reversion, high volatility.

of hours (1 day, 1 week, 1 month respectively), as can be seen in figures 13 and 14.

Although not directly comparable as we are fundamentally dealing with different underlying parameters, there exists an overarching concept: Maintain parity through economically controlled incentive mechanisms - yet not at the expense of users/supply growth. As such, we see here that under the same (median) simulation assumptions, the option box not only performs better as it relates to raw pct/supply redeemed (reduction), it provides users with sufficient time in the form of realizing a net incentive to repay debt cheaper (and/or coupled with external algorithmically controlled rates/staked ETH perf. fee cuts as a function of eBTC $pAMM_t$). In particular, given the undeniable market demand to loop/leverage staked ETH in a "var yield (var performance fee) $\leftrightarrow$ zero interest rate" economy, this could find itself carrying importance as eBTC $->$ 1-ϵ w.o. hard redemptions.

That being said, there are a few skewed assumptions as it relates to the raw results. For one, we assumed that hard and soft redemption "arbitrage" opportunities provided equivalent demand when eBTC $>$ 98 cents, of which is objectively not the case (i.e. quick redemption that in theory could be accomplished with a simple flashloan vs locking capital for T days, of which additionally exposes one to BTC directional risk for a given amount of time). As such, this could result in the market forcibly pricing the realized "soft floor" differently, especially during times of volatility, as

$$\begin{aligned} \textit{net redeemer profit} \approx\ & max(pAMM_T, strike) - \\ & \textit{redeemer realized execution/price} \pm \%\Delta pBTC/USD_{T-0} \end{aligned}$$

However, as we mentioned earlier, if we assume the majority of eBTC liquidity follows the utilization of the Curve stableswap invariant, pAMM fundamentally requires much less raw liquidity to revert to mean (as it relates to net redeemer trade execution, gradually decreasing as eBTC $->$ 0), and thus in turn can be relied upon to dictate system parameters.

4.2 Final Thoughts

Ultimately the integration of the option box can be seen as free lunch in theory, however the question arises as to how users will react to this system right off the bat, thus we recommend starting off with a normalized iteration of Liquity's redemption mechanism (whilst keeping the option box on the backburner if applicable at a later date).

In this iteration, we recommend initially setting α to 1 (double that of LUSD) to account for eBTC's higher redemption base fee (as a result of max oracle deviation) and the overall effect of Curve liquidity when 1 - ϵ $>$ $pAMM$ (as mentioned earlier). For instance, since we assume the price to revert to $> 1 - \epsilon$ through less liquidity utilization/redemptions, so too we can (dis)incentivize accordingly, s.t. redemptions are mitigated to a larger extent whilst still maintaining a hard price floor (and theoretically allocating more time for troves to close/top up their looped positions and/or repay debt cheaper).

These adjustments are strategically done as opposed to Liquity's flat given its objective immutable properties, whereas these parameters can theoretically be altered by Badger governance at a later date if needed. Note that LUSD redemptions simply follow the QTM (quantity theory of money), which states that a k% supply reduction results in a k% price increase in the underlying, i.e. obviously nothing relating to Curve slippage assumptions. Here, we are effectively creating a limiting property akin to that of a fixed upperbounded daily redemption limit (maintaining exponential decay with a 12 hr half life, etc), however the Curve pool further indirectly mitigates redemption speed (as it relates to pAMM growth) due to lower redemption price.

5 Simulation code

The implementation code is publicly available on Github, under MIT license.

- The backend code to execute the simulation of liquidation events is available here.
- The backend code to execute the simulation of the redemption events is available here.
- The relevant commit hash for both files is 9134d23894fa33c5b0c49a0b31a751999bb69bff.
- The front end that browse the liquidation simulation results is implemented as a react app and available here.

References

[1] *AAVE Protocol Whitepaper.* URL: https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf. (accessed: 9.06.2021).

[2] The Risk DAO. *Simulation results.* URL: https://tinyurl.com/yns6rpxu.

[3] The Risk DAO. *Simulation results.* URL: https://tinyurl.com/4hyj8ezv.

[4] The Risk DAO. *Simulation results.* URL: https://tinyurl.com/3csd5n96.

[5] Michael Egorov. *StableSwap - efficient mechanism for Stablecoin liquidity.* URL: https://classic.curve.fi/files/stableswap-paper.pdf. (accessed: 8.05.2023).

[6] *Euler Finance White Paper.* URL: https://docs.euler.finance/getting-started/white-paper. (accessed: 3.05.2023).

[7] Richard Pardoe Robert Lauko. *Liquity: Decenteralized Borrowing Protocol.* URL: https://docsend.com/view/bwiczmy. (accessed: 26.05.2022).

[8] Geoffrey Hayes Robert Leshner. *Compound: The Money Market Protocol.* URL: https://compound.finance/documents/Compound.Whitepaper.pdf. (accessed: 02.03.2020).