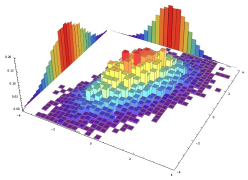
Y2K Economic Audit

SmolQuants

December 2022



Contents

1 Introduction

SmolQuants was engaged by Y2K to review the economic mechanisms behind their new Earthquake bonds over the course of four weeks. The audit focused on providing cost of attack figures for manipulating the MIM price oracle to trigger depeg events on the associated Y2K market and the manner in which the Y2K protocol prices its Earthquake bonds.

2 Background

Y2K is a protocol for structured products on exotic peg derivatives. The protocol allows participants to hedge or speculate on the depeg risk of a pegged asset.

With Earthquake vaults, Y2K applies the notion of a catastrophe bond to DeFi for depeg events on stablecoins, liquid wrappers and other derivative products. Users can buy/sell insurance against the event of the price of the underlying going below a protocol-set strike threshold. The catastrophe bonds can be viewed as binary option markets on depeg events for stablecoins.

The protocol currently offers the following binary put options with strike prices K paired against USD:

- MIM: `K = 0.9759`
- USDC: `K = 0.9979`
- USDT: `K = 0.9919`
- FRAX: `K = 0.9909`
- DAI: `K = 0.9969`

Earthquake has two sets of vaults, hedge and risk, that buyers and sellers of the option, respectively, can deposit capital into during an initial deposit period. The deposit phase is where price discovery happens for the value of the Y2K binary put on the stablecoin depeg.

Once deposited, users are *not* able to withdraw their capital until either after expiry or once a depeg event is triggered (i.e. price goes below strike K).

3 Structure

The Y2K economic audit is broken into two sections:

- Oracle manipulation
- Insurance pricing mechanism

The first section deals with cost of attack calculations for manipulating the underlying Curve MIM pool to intentionally trigger depeg events on Y2K.

The second section focuses on the current insurance pricing mechanisms implemented by the protocol. It dives deeper into ways to improve pricing to mitigate the issues mentioned.

4 Oracle Manipulation

MIM is likely the riskiest of all markets offered, reflected in the lower strike price set by the protocol. Focus for the oracle manipulation analysis was on MIM given the increased risks to option sellers of this market.

The analysis around Curve pool manipulation took a conservative approach and assumed the worst-case scenario of the Chainlink oracle relaying price directly from the Curve pool (but no flashloan attacks), as Chainlink docs can be somewhat opaque with respect to the MIM feed.

4.1 Summary

- ~62% of MIM circulating supply and the majority of liquidity for the token is in the MIM Curve metapool. Meaning, price discovery should occur solely through this lone pool, making it an easy target for price manipulation
- An attacker could purchase a Y2K put on MIM, mint MIM through Abracadabra, sell into the Curve pool to trigger a depeg, collect on Y2K, swap back through the Curve pool and repay the MIM loan
- Upfront capital to execute this attack requires a ~47M MIM mint, given current liquidity conditions
- Slippage lost on Curve, however, would only be ~230K USD
- The attack is **not** currently possible due to borrow caps on Abracadabra being fully exhausted
- Y2K should consider implementing caps on risk vault deposits at or near this Curve slippage loss amount to completely eliminate the profitability of the attack in the event Abracadabra increases new borrows in the future

4.2 Issues

4.2.1 Risks with the MIM Curve Metapool

Users of Abracadbra can mint MIM through a CDP mechanism, with the loan backed by various collateral types. Analytics dashboards for MIM are provided by Abracadabra breaking down outstanding supply, borrows, collateral, and collateral profile.

An overview as of 2022-12-04,

- ~85M MIM of total borrows (i.e. MIM circulating supply)
- ~154M USD worth of collateral backing

with the majority of activity (~97.5%) on Ethereum mainnet.

However the top holders of MIM on Ethereum are heavily concentrated in a few addresses, with **~62% of the circulating supply in the MIM Curve Metapool**.

To understand the liquidity distirbution of MIM, several relevant top holders from Etherscan's MIM Token page are listed below:

1. 472.016M in anyMIM contract
2. 115.069M in a CauldronOwner contract
3. 53.217M in Curve Metapool with MIM and 3Crv
4. 30.724M in Abracadabra Multisig
5. 5.892M in Abracadabra Degenbox
6. 1.290M in a single EOA
7. 0.795M in Sushiswap BentoBox V1
8. 0.688M in Gemini 4
9. 0.200M in Bitfinex: Hot Wallet
10. 0.183M in SushiSwap: MIM 2 Pool (MIM/WETH)
11. 0.022M in Uni V3: MIM-USDC Pool

As the majority of the liquidity for MIM lies in the single Curve metapool, price discovery for MIM vs USD will likely happen through this pool. This is a significant risk for Y2K when offering markets on MIM, as price manipulation of this lone pool by a large MIM holder will likely be difficult to arbitrage back due to insignificant liquidity on other major DEXs and CEXs. This should also affect *any* oracle reporting MIM price due to the significant concentration of circulating supply in the Curve pool, regardless of the buffers put in place.

4.2.2 Manipulating the Curve Pool to Trigger Depegs

4.2.2.1 Curve Math Curve metapools are pools paired with an underlying base pool's LP token. In the case of MIM, the base pool is the Curve 3pool composed of DAI, USDC, and USDT.

Some notes on Curve V1 pools are provided here for reference. The general takeaway is the Curve pool acts like a superposition of a constant product pool $\prod_i x_i = (D/n)^n$ with a constant sum pool $\sum_i x_i = D$.

The differences with constant product are most extreme near the equilibrium point of balanced reserves (i.e. price = 1), where the marginal price curve $P_{ij} = -dx_i/dx_j$ flattens out

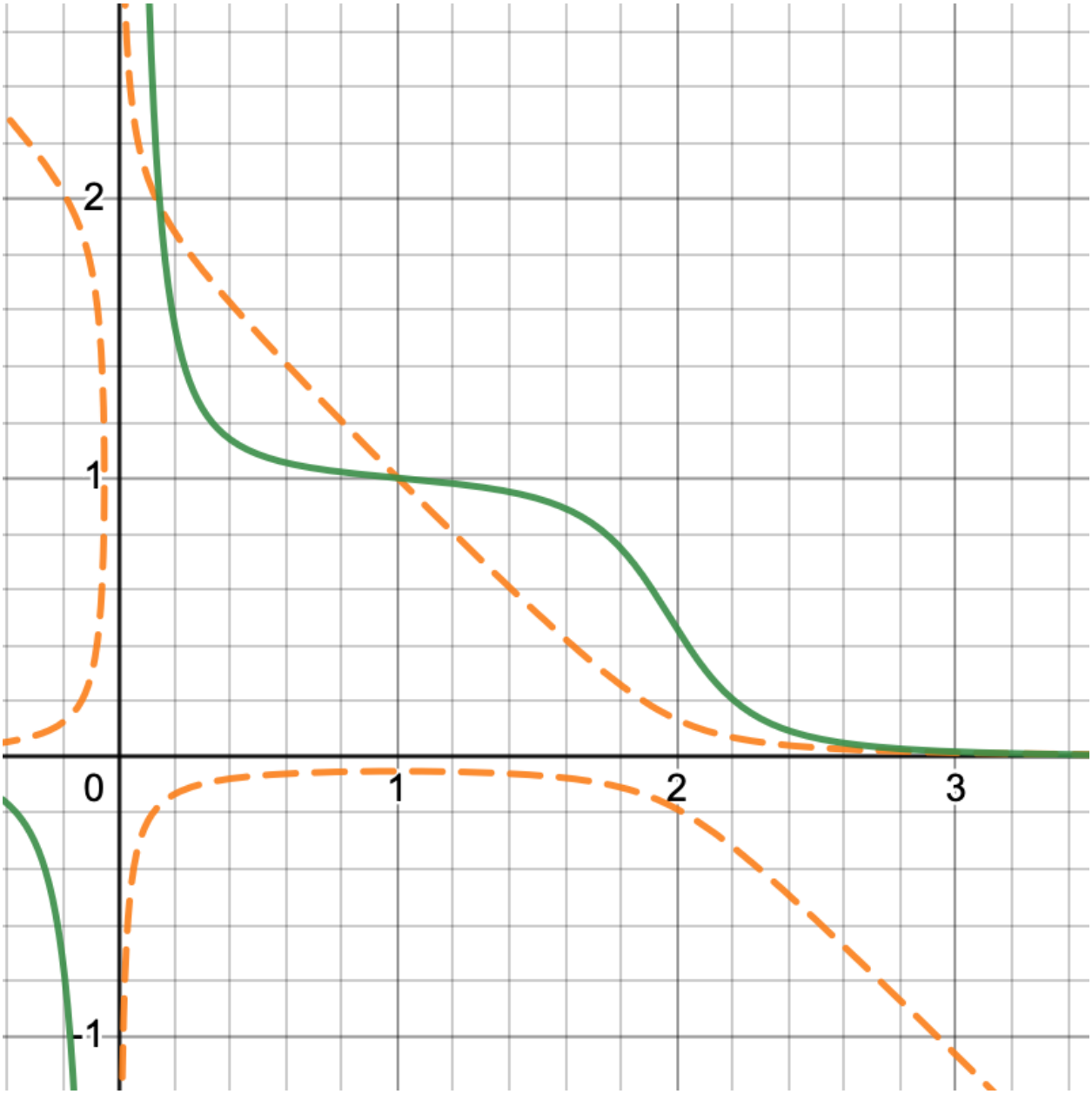


Figure 1: StableSwap price chart

Green is marginal price $P(x)$ as a function of x reserves, orange-dotted is the Curve invariant plotted with respect to coin balances (x, y). Only looking at the two-coin case for simplicity.

The tradeoff made by the Curve pool is significantly less slippage when reserves are balanced, but extreme slippage once significant imbalance occurs (i.e. price goes to zero rapidly). This is easiest to see from the marginal slippage chart $S_{ij} = -dP_{ij}/dx_j$

Red is marginal slippage $S(x)$ as a function of x reserves, green-dotted is the marginal price, orange-dotted is

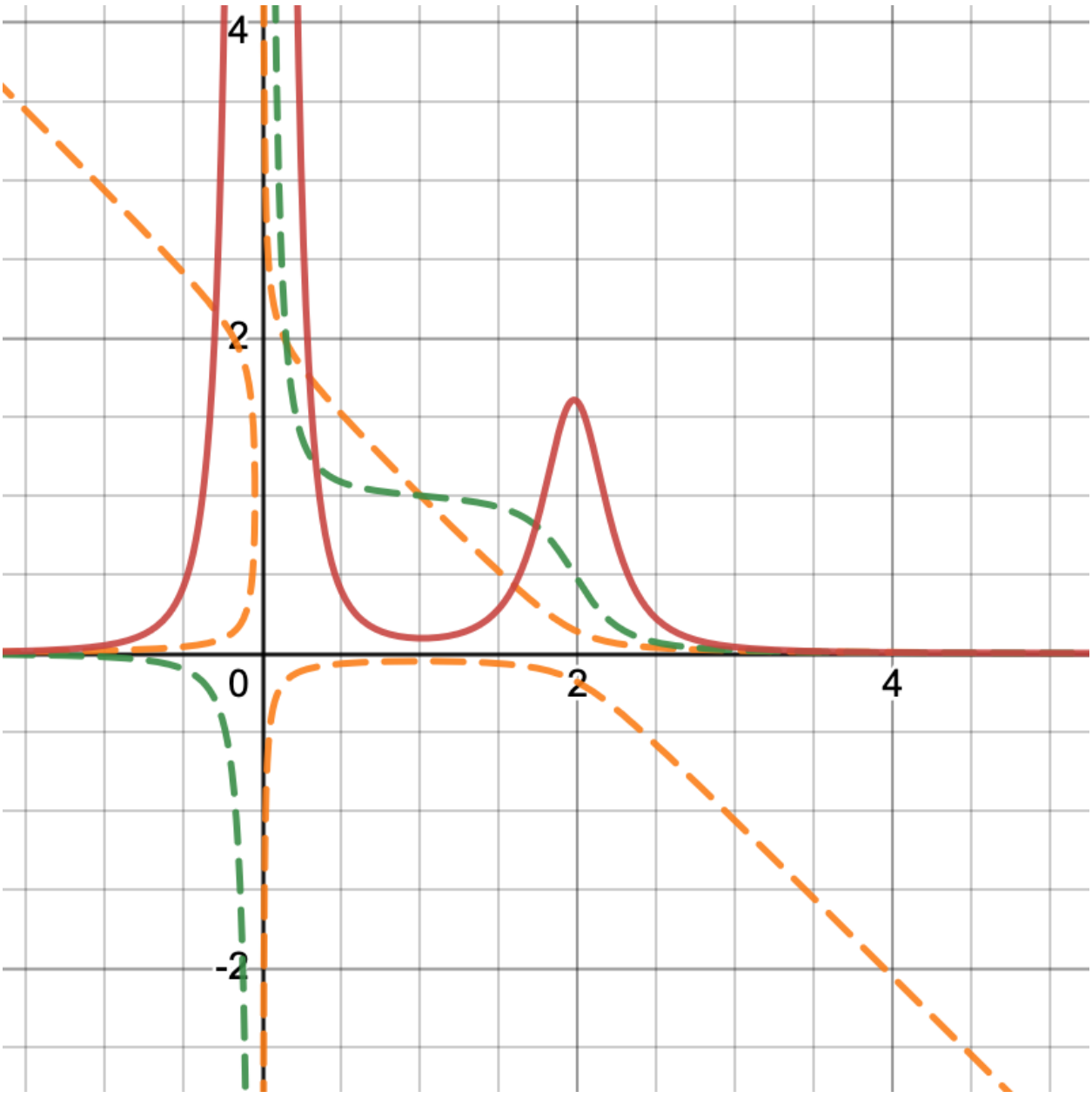


Figure 2: StableSwap slippage chart

the Curve invariant.

Slippage is near zero when balanced, but increases rapidly as imbalance occurs.

4.2.2.2 Attacking The Curve Pool An attacker could use the MIM Curve metapool to manipulate the price of MIM vs USD in their favor, so as to trigger a depeg event on the MIM Y2K binary put oracle. The oracle manipulation attack goes as follows:

1. Purchase binary put via Y2K on MIM
2. Wait until Y2K vault deposits close
3. Mint MIM via Abracadabra CDP
4. Sell MIM for USDC/USDT/DAI through the MIM Curve metapool
5. Oracle reports price below strike `K` due to sell, triggering depeg event on Y2K
6. Claim insurance payout via Y2K
7. Sell USDC/USDT/DAI for MIM through same MIM Curve metapool
8. Repay MIM loan from Abracadabra

The PnL for this attack is

```
PnL for attack = Y2K payout - Y2K premium - Slippage on Curve
```

where the calculation for how much the attacker loses to slippage on Curve uses the math from the prior section. Using parameters from the MIM Curve metapool in this plot as an estimate roughly shows the pool breaks below the Y2K strike price when MIM makes up ~96.5% of the pool balances.

4.2.2.3 Cost of Attack Cost of attack for manipulating the MIM Curve metapool is the minimum amount of upfront MIM capital needed to sell into the pool for the price on Curve to dip below the Y2K strike of `K=0.9759`.

This can be found by inverting the marginal price function $x = P^{-1}(p)$ and taking the delta between the current price P_0 and the target strike price K

$$\Delta x = P^{-1}(K) - P^{-1}(P_0)$$

where x is in units of MIM.

Since Curve math isn't super nice to work with, an easier approach taken in the provided oracle manipulation notebook is to simply try many values for Δx in the known function $P(x + \Delta x)$ until the output is very near to the strike price.

To be even more rigorous, the provided oracle manipulation script deploys in mainnet-fork a new mock Curve V1 pool with the current MIM metapool parameters. Mock tokens are minted to this pool to replicate the current liquidity conditions in the actual metapool. The script generates in a csv file the results of executing the attack against the mock pool for various input sizes sold into the pool. To run, execute from the base directory

```
hatch run ape run curve_manipulation
```

As of 2022-12-04,

- **~47M MIM mint via Abracadabra** is required to manipulate the Curve pool to the strike price of 0.9759
- **>52M USD of upfront collateral for CDP** assuming a maximum collateral ratio of 90% for cauldron of choice
- **~230K MIM is lost to slippage** from executing the attack

Any payout from Y2K less fees and premium that is greater than ~230K USD will make this attack worthwhile to try. Though, there are inherent risks in whether or not the Chainlink oracle relays the price post-swap.

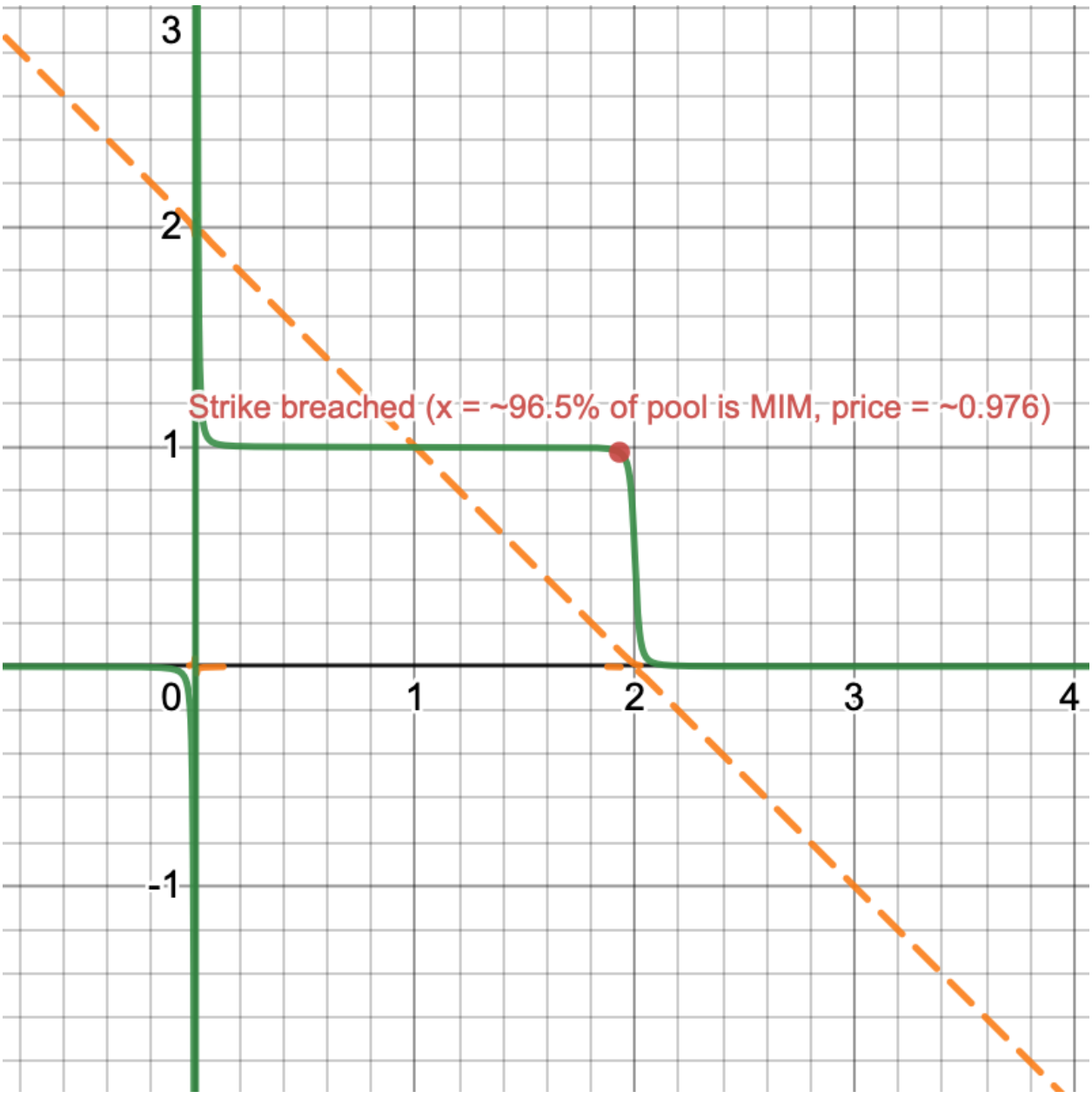


Figure 3: StableSwap depeg chart

4.3 Recommendations

Y2K MIM risk vaults currently have deposits of > $1.678M, so this attack would be potentially profitable if not for borrow caps on Abracadabra being fully exhausted.

Mitigations that Y2K should consider are below.

4.3.1 Caps on Y2K Risk Vault Deposits

Caps on Y2K risk vault deposits as a function of current liquidity in the Curve pool would enforce PnL < 0 for this attack by limiting the maximum payout the attacker would receive from Y2K. Conservatively, the cap should be set to the expected slippage loss on Curve when executing the attack. Given current conditions, this risk vault cap on MIM would be rather low, however, at ~230K USD.

4.3.2 Monitor CDP Caps on Abracadabra

CDP caps on Abracadabra for minting new MIM should be constantly monitored when offering MIM binary puts. The possibility of new large loans being offered by Abracadabra increases the risk of this attack being executed profitably.

4.3.3 Price Insurance via Bonding Curve

The Y2K premium pricing mechanism does not rely on a bonding curve. Therefore, there is no explicit slippage mechanism to deter larger players from buying greater pro-rata rights to the risk vault deposits in the event of a depeg. Increased slippage for greater size would deter this manipulation attack by eating into the Y2K portion of the payout in the PnL expression above. The insurance pricing analysis goes into further detail for the Y2K premium pricing mechanisms.

5 Insurance Pricing Mechanism

Expressions for the insurance pricing mechanism analysis follow those of the Y2K whitepaper with additions:

- $B = \sum_i B_i$ is the total premiums paid by hedge vault depositors
- $S = \sum_j S_j$ is the total collateral posted by risk vault depositors

Protocol fees are ignored for the sake of simplicity. Overview of the Y2K payout structure is below.

During the deposit phase,

- Buyer i pays B_i in premiums by depositing to the hedge vault
- Seller j risks S_j in collateral by depositing to the risk vault

If no depeg happens prior to expiry,

- Buyer i receives zero payout
- Seller j receives a pro-rata share of hedge vault premiums plus their original risk collateral back: $(S_j/S) \cdot B + S_j$

In the event of a depeg,

- Buyer i receives a pro-rata share of risk vault collateral: $(B_i/B) \cdot S$
- Seller j receives a pro-rata share of hedge vault premiums: $(S_j/S) \cdot B$

5.1 Summary

- The pro-rata payout structure of the Y2K binary put causes issues from a price discovery and contract size perspective
- Risk vault depositors are effectively quoting an ask price of 0 (accept any price from hedgers) at the start of the deposit period, when no hedgers have bought insurance yet

- Hedge vault depositors do not know how much size they are actually covered for at time of purchase, which is an issue if hedgers are using the insurance as a hedge on a fixed stablecoin portfolio size
- Y2K should consider implementing a binary option AMM for price discovery, where minters of the risk token (binary put) provide liquidity post-mint and hedgers buy the newly minted token from the AMM

5.2 Issues

5.2.1 Quoting the Put with Zero Hedgers

There is a significant mispricing of risk when the vaults first open due to the manner in which price discovery happens on Y2K. The first buyer $i = 0$ can bid whatever amount B_0 they want for the full rights to *all* of the collateral in the risk vault S in the event of a depeg. The initial round of sellers are forced to sell at this price chosen by the first bidder given the mechanics of the protocol.

A two-player example to illustrate:

- Seller deposits $1M of ETH in the risk vault to underwrite the depeg insurance
- Buyer deposits $0.01 of ETH in the hedge vault to buy the depeg insurance *on $1M payout*
- If depeg occurs, buyer receives $1M and seller receives $0.01.

As the first binary put buyer is able to specify their own price through depositing whatever amount B_0 they desire, the initial round of sellers depositing into the risk vault are effectively quoting the ask for the binary put at a price of 0. It is unlikely risk sellers actually are pricing the binary put at this value.

To prevent this scenario, sellers of the binary put should be able to specify an initial price they're willing to sell at, similar to the price specified when e.g. initializing a Uniswap pool.

5.2.1.1 Pricing the Binary Put

To dig a bit deeper into what quoting the ask implies, note that pricing the binary put is effectively pricing the probability of a depeg event before expiry.

For simplicity, assume the puts are European (i.e. aren't triggered prior to expiry). The market value for the binary put can be expressed as the discounted expectation of the future payoff under the risk-neutral measure Q:

$$V(\tau) = Se^{-r\tau} \cdot \mathbb{E}_Q[\mathbf{1}_{P_T \leq K}] = Se^{-r\tau} \cdot \mathbb{P}_Q[P_T \leq K]$$

where

- P_t is the price of the underlying stablecoin v.s. USD at time t
- τ is time to expiry
- T is expiry time
- r is the risk-free rate
- S is the total collateral deposited in the risk vault
- $\mathbf{1}_{P_T \leq K}$ is the indicator function

What Y2K vault depositors are trading when buying/selling the option $V(\tau)$ is the probability of the depeg event $\mathbb{P}_Q[P_T \leq K]$.

If sellers must honor an initial hedge vault depositor paying $B_0 \to 0$ with no other hedgers coming in after, sellers are forced into expressing the view that the probability of a depeg event occurring within the epoch must be zero $\mathbb{P}_Q[P_T \leq K] \to 0$, simply due to the initial hedge vault depositor bidding a low price of 0. This is particularly the case given vault depositors *cannot* withdraw their capital once deposited during the deposit period.

5.2.2 Size Not Known at Time of Purchase

The amount of insurance purchased (payout size) by the hedge vault depositor i with B_i of collateral is variable and depends on the other hedge buyers that may come in after the purchaser $k > i$ given the pro-rata

payout structure. Therefore, the size the buyer is covered for is not actually known to the buyer at the time of purchase. Usually with insurance products (and options), the purchaser of the coverage buys the contracts for a known fixed coverage size.

A three-player example to illustrate:

- Seller deposits $1M of ETH in the risk vault to underwrite the depeg insurance
- Buyer 1 deposits $10 of ETH in the hedge vault to buy depeg insurance
- Buyer 2 deposits $90 of ETH in the hedge vault after buyer 1 to also buy depeg insurance
- If depeg occurs, buyer 1 receives $100K and buyer 2 receives $900K.
- If buyer 2 had not purchased insurance after buyer 1, buyer 1 would have received $1M.

This becomes a real problem when a trader uses the purchased insurance as a hedge for a fixed amount of stablecoins held in their portfolio, as the hedge has been significantly reduced due to demand from other buyers purchasing *after* them – from $1M to $100K in the case of buyer 1 in the three-player example.

Another way to realize this is by examining the pro-rata depeg event payout $(B_i/B) \cdot S$. Per-unit of risk vault collateral for the payout should be the number of contracts buyer i purchased of the binary put:

$$\text{OI} = \frac{B_i}{\sum_k B_k}$$

The sum $\sum_k B_k$ in the denominator of the open interest expression increases the more other buyers buy insurance. Therefore, the open interest of buyer i decreases significantly the more insurance is purchased by other buyers after i. Ideally, i should be able to purchase a fixed amount of open interest/coverage tokens that represent a fixed portion of the claim on the total payout amount S. The initial purchase price for the option would need to be set higher than a starting ask price of 0 for this to work (see Quoting the Put with Zero Hedgers).

5.3 Recommendations

5.3.1 Y2K Binary Option AMM

Consider relying on an AMM with a price curve for the next iteration of the protocol. Y2K does currently enable price discovery for the binary puts during the deposit period, but it's difficult to overcome the issues above when taking the current pro-rata approach to price discovery.

A helpful reference example may be the Squeeth approach, where option sellers collateralize and mint the derivative token separately from the act of actually selling the option to buyers. Instead, selling occurs through sellers providing liquidity for the minted option token on Uniswap vs ETH. The derivative vs ETH Uniswap pool is initialized with a suitable initial price by the first minters.

A possible alternative to the Squeeth approach of piggybacking on Uniswap to make a market for the binary put would be a specialized Y2K AMM for the binary option. Risk vault depositors would collateralize and mint the derivative token for a given fixed size, then subsequently provide liquidity vs ETH to the specialized AMM for buyers to then purchase insurance from the liquidity pool (eliminates the hedge vault). The Y2K price curve should ultimately look similar in shape to the expected CDF (bound between 0 and 1 per unit of risk vault collateral), as this is what traders are trading. Pool initializers must also be able to set a suitable initial price at pool deployment.