Secure Finance: Simulation and Risk Assessment

The CEL Team

Executive Summary

What is this Document About?

This document presents a simulation-based risk analysis for a Filecoin-collateralized stablecoin called USDFC. It adapts smart contract logic from Liquity [1] and MakerDAO-like mechanisms and then examines how well these concepts hold up on the Filecoin network.

Why Does it Matter?

USDFC's stability depends on managing three core risks:

- *Collateral Risk* (e.g., large drops in FIL price),
- *Redemption Risk* (forced removal of a user's collateral),
- *Liquidity and Peg Stability* (ensuring USDFC stays near $1).

Key Findings and Output

- **Collateral Ratios:** We implement a robust simulation environment to stress-test the robustness of USDFC's mechanism under several different market conditions.
- **Collateral Ratios:** A 110% overcollateralization is sufficient to avoid mis-liquidation risk.
- **Redemption Mitigations:** Defensive structures such as buffer troves divert forced redemptions away from users' own collateral, mitigating forced-sale concerns.
- **System Robustness:** Under moderate volatility and appropriate parameters, the stablecoin price remains quite close to peg. Stress tests (90% FIL price crashes) identify potential shortfalls if the system lacks liquidity or uses too-low collateralization thresholds.

Who Should Read This?

Protocol Designers, Investors, Liquidity Providers, and anyone with an interest in stablecoin risk analysis.

Guide to This Document

I. **Background and Glossary** (Section 1): Definitions of the key terms (e.g., troves, collateral ratios) and a high-level explanation of how the USDFC protocol works.

II. **Simulation Methodology**(Section 3): Describes how we model FIL price, trove liquidations, redemptions, and other system variables.

III. **Risk Assessment** (Sections 4 and 6): Examines liquidation probability and redemption events under various market conditions.

IV. **Scenario Analysis and Mitigation** (Section 5): Explores different stress scenarios (crashes, liquidity shortages) and risk-mitigation strategies (buffer troves).

V. **Conclusions and Recommendations** (Section 8): Key takeaways for implementing a safe USDFC ecosystem on Filecoin.

VI. **Appendix** (Section A): Pseudocode, additional data tables, and details for replicating or extending the simulations.

1 Background and Glossary

1.1 Stablecoins and Collateralization

A **stablecoin** is a cryptocurrency pegged to a stable asset—often the U.S. dollar. Many stablecoins maintain their peg by holding collateral. For example, MakerDAO's DAI is backed by ETH, and Liquity's LUSD is backed by ETH as well. USDFC, the protocol discussed here, is a *CDP-style stablecoin* on the Filecoin (FIL) network.

1.2 Key Terms and Definitions

Trove A user's collateralized debt position. The user deposits FIL as collateral and mints USDFC stablecoins.

Collateral Ratio (CR) The ratio of a trove's total collateral value to its debt. For example, if a user locks $150 worth of FIL and mints 100 USDFC, the CR is 150%.

Liquidation If the collateral ratio falls below a minimum threshold (e.g., 110%), the trove is liquidated. The collateral is seized and sold (or used to reimburse the system).

Redemption A forced purchase of collateral by someone who returns USDFC to the protocol. Troves with the lowest CR are targeted first; this mechanism helps the stablecoin hold its $1 peg.

Buffer Trove An additional trove set at a lower CR than the user's main trove, deliberately making it a more attractive target for redemptions.

Liquidity Pool (LP) On-chain AMMs that facilitate USDFC-FIL swaps, affecting the stablecoin peg stability.

1.3 Why Filecoin?

Filecoin (FIL) has unique properties (e.g., an active storage market, a different adoption curve than ETH) that can lead to different price dynamics. Adapting Liquity's or MakerDAO's model to FIL might work, but we must verify with simulation whether parameters like the minimum CR or redemption fees require adjustment.

Remark 1 (Real-World Context)**.** *In Ethereum-based stablecoins (MakerDAO, Liquity), we have seen how heavy market volatility or congested blocks can trigger mass liquidations. Filecoin's block times and liquidity may differ from ETH, so risk parameters cannot be copied blindly.*

2 Introduction

USDFC [4] is a CDP-style stablecoin designed for native deployment on Filecoin. It is built on smart contract logic ported from the Liquity protocol [1], and its system design and parameter settings—such as the minimum collateralization ratio—are optimized based on values originally established during Ethereum's launch in 2018. Given this, the main goal of this work is to understand *how well* can this model be ported to the Filecoin network.

Specifically, this technical analysis will focus three main tasks:

1. Developing a simulation environment for a FIL-backed USDFC

2. Using this simulation environment to evaluate protocol risk from the perspectives of Liquidity Providers, Market Makers, and stablecoin holders. It will also involve sensitivity analyses to determine how variations in protocol parameters affect both the stablecoin system and the broader Filecoin economy. Additionally, the study will include economic stress testing to identify potential failure modes and establish safe operating bounds for the USDFC protocol.

3. Understand redemption risk

The mechanism behind USDFC can be found in [1, 4]

3 Model Architecture

TL;DR. *We create a robust simulation environment extending the work of [2]. Our methodology allows us to track several quantities over time, as well as functions and statistical estimators of them.*

We follow an approach similar to that of [2], however, as we will discuss shortly, we present several differences in our implementation that result in a more robust, uncertainty-aware simulator.

The simulation employs a Monte Carlo framework to model the USDFC stablecoin system dynamics over discrete time steps. The main quantities used in this Section are shown in Table 1 below. The implemented code can be found on GitHub[1]. The advantage of this methodology is that it allows us to experiment different market conditions, counterfactual scenarios, etc, while at the same time providing uncertainty bounds on the desired quantities of interest. These uncertainties arise from the aleatoric nature of the whole system. Indeed, components such as FIL price, redemption frequency, amounts, etc are, in reality, random processes.

Remark 2 (On Simulations)**.** *While our methodology can be understood as a sort of digital twin (in the sense that it utilizes the same mechanisms as in [1, 4], the behavior of the derived, observable quantities will depend on several factors, including initial conditions, etc. These initial conditions are, in turn, impossible to capture with 100% accuracy. As such, we emphasize that the current results are informative in a* directional *sense.*

Price trajectories for FIL and SFTOKEN tokens are generated via Geometric Brownian Motion (GBM), represented by the equation:

$$S(t) = S_0 \cdot \exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)\right), \tag{1}$$

[1] https://github.com/juanpmcianci/SF

Variable	Definition
$S(t)$	Price at time t following GBM: $S(t) = S_0 \exp\Big((\mu - \frac{\sigma^2}{2})t + \sigma W(t)\Big)$
S_0	Initial price (applied to both FIL and LQTY simulations)
μ	Drift coefficient in the GBM process
σ	Volatility coefficient in the GBM process
$W(t)$	Cumulative sum of standard normal shocks scaled by $\sqrt{dt}$
Q_{FIL}	Collateral amount in FIL tokens held in a trove
D	USDFC supply (or debt) associated with a trove
$\text{CR}_{\text{current}}$	Current collateral ratio: $\frac{P_{\text{FIL}} \times Q_{\text{FIL}}}{D}$
S_{pool}	Stability pool amount used for absorbing liquidations
L	Liquidity pool amount that influences the USDFC price
P_{USDFC}	Price of the USDFC stablecoin
θ	Exponent in the stability pool update equation
δ	Scaling parameter in the USDFC price update based on liquidity
r_{return}	Return from liquidations and airdrop gains
r_{natural}	Natural rate parameter for stability adjustments

Table 1: Definitions of key variables used in the simulation.

where S_0 is the initial price, μ is the drift, σ is the volatility, and $W(t)$ denotes the cumulative sum of scaled standard normal shocks.

Troves are modeled as collateralized positions characterized by a FIL collateral amount Q_{FIL} and a minted USDFC supply D. The current collateral ratio (CR) is given by:

$$\text{CR}_{\text{current}} = \frac{P_{\text{FIL}} \times Q_{\text{FIL}}}{D}. \tag{2}$$

A trove is liquidated if its $\text{CR}_{\text{current}}$ falls below a threshold (e.g., 110% over collateralization). The liquidation gain is computed as the difference between the market value of the collateral and the debt, potentially scaled by the stability pool's size.

The simulation also incorporates dynamic operations on troves. New troves are opened by sampling a target collateral ratio from a chi-square distribution and determining the collateral quantity using a gamma distribution. Adjustments are made when the deviation from the initial collateral ratio exceeds preset bounds, leading to modifications in the trove's debt or collateral. Associated issuance fees are computed as a function of the change in the USDFC supply.

Protocol stability is maintained via a stability pool updated according to:

$$S_{\text{pool},t} = S_{\text{pool},t-1} \times d \times (1 + \epsilon_t) \times (1 + r_{\text{return}} - r_{\text{natural},t})^{\theta}, \tag{3}$$

where d is a drift factor, ϵ_t is a stochastic shock, r_{return} is the return from liquidations and airdrops, and $r_{\text{natural},t}$ is the natural rate at time t.

Furthermore, the USDFC price is recalculated using liquidity adjustments:

$$P_{\text{USDFC},t} = P_{\text{USDFC},t-1} \left(\frac{L_t}{L_{t+1}} \right)^{1/\delta}, \tag{4}$$

with L_t and L_{t+1} representing the liquidity pool at consecutive time steps, and δ being a scaling parameter. Arbitrage mechanisms further adjust the system: if $P_{\text{USDFC},t}$ exceeds an upper bound, new troves are opened; if it falls below a lower bound, redemption processes are triggered. The main simulation loop is shown in Algorithm 1. The relevant sub-routines are shown in the Appendix.

Algorithm 1 USDsf Stability Simulation Main Loop

Input: Simulation parameters (GBM parameters, fee rates, thresholds, etc.)
Initialize:
- Generate initial FIL price $S_{0,\text{FIL}}$ and LQTY price $S_{0,\text{LQTY}}$
- Initialize troves, stability pool S_{pool}, liquidity pool L, and other state variables.

GeneratePricePath(FIL) ▷ Using GBM for FIL
GeneratePricePath(LQTY) ▷ Using GBM for LQTY
for $t \leftarrow 1$ **to** T **do**
    Update current FIL price $P_{\text{FIL},t}$
    $[troves, \text{fees}, \ldots] \leftarrow$ UpdateTroves(troves, t, $P_{\text{FIL},t}$, $P_{\text{USDsf},t-1}$)
    $S_{\text{pool},t} \leftarrow$ UpdateStabilityPool($S_{\text{pool},t-1}$, t, return_stability, D_{total})
    $[L_{t+1}, P_{\text{USDsf},t}] \leftarrow$ UpdateLiquidityAndPrice(L_t, $P_{\text{USDsf},t-1}$, troves, t)
    $[troves, P_{\text{USDsf},t}] \leftarrow$ ArbitrageAdjustments(troves, t, $P_{\text{USDsf},t}$)
    $[P_{\text{LQTY},t}, \ldots] \leftarrow$ UpdateLQTYMarket(data, t, $P_{\text{LQTY},t-1}$)
    Record all state variables at time t.
end for
Output: Time-series data, total liquidations, final liquidity, and final USDsf price.

Our simulation environment is able to simulate and track the following observable quantities $\{\theta_t,\ t \geq 0\}$ for arbitrary simulation lengths.

It is also able to estimate arbitrary functions f of these observables, of the form:

$$\mathsf{QoI}(\theta_t) = \mathbb{E}_\nu[f(\theta_t)] = \int f(\theta_t)\nu(\mathrm{d}\theta_t, \mathrm{d}t), \tag{5}$$

with ν the probability measure induced by the simulation. Less abstractly, Equation (5) can translate into questions such as *what is the average number of open troves in a day? What is the probability of being liquidated for a given CR? What is the expected number of redemptions?*

Variable (θ_t)	Definition
$S_{\text{FIL}}(t)$	Price of FIL over time
$N_{\text{open}}(t)$	Number of open troves over time
$N_{\text{close}}(t)$	Number of closed troves over time
$N_{\text{liq}}(t)$	Number of liquidations
$N_{\text{red}}(t)$	Number of redemptions
$L(t)$	Pool liquidity
$\mathsf{RP}(t)$	Redeemed amount
$\mathsf{C}(t)$	USDFC supply

Table 2: Observable variables.

3.1 Example Run

We run our simulation engine for a year forth of simulated data (on an hourly basis), on various market conditions, as described below. As we can see, our simulation is able to capture the stochastic behaviour of these type of systems, as well as the effect that different market conditions might have on them. Results are shown in Figures 1 through 4. Many additional results can be found in [3]. In those experiments, we simulate the expected behaviour of the quantities on Table 2 for 4 different market scenarios (*base,bullish,bearish* and *high volatility*).

Specifically, in Figure 1 we observe different potential price trajectories for different market conditions for FIL. These market conditions are expressed in terms of drift and volatility (μ, σ above) and are obtained using historical data based on recent market conditions [2]

[2]Notice that at the time of writing (March 2025) the market or FIL was not doing too well.

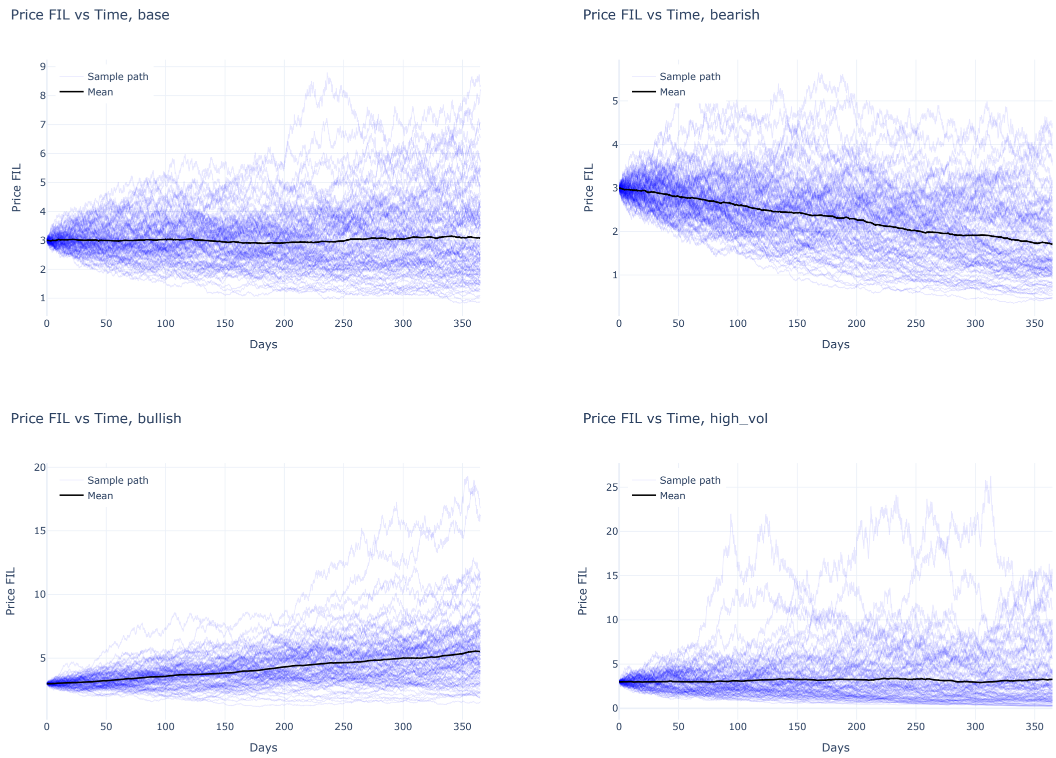

Figure 1: Different price-paths realisations for FIL under different market conditions; *base* (top left), *bearish* (top right), *bullish* (bottom left) and *high-volatility* (bottom right).

In figure 2 we see the corresponding USDFC price trajectories for the abovementioned market conditions. As we can see, the protocol shows that the asset is stable across a variety of market scenarios; however, its uncertainty bounds are ensitive to periods of high volatility.

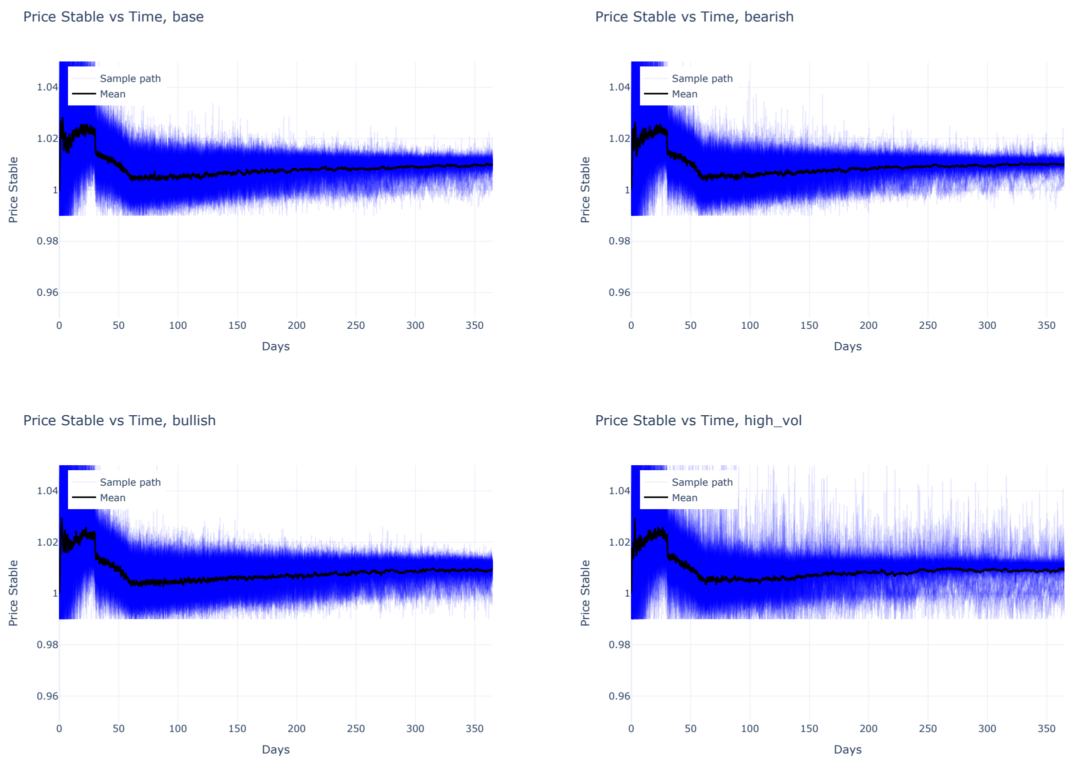

Figure 2: Different price-paths realisations for USDFC under different market conditions; *base* (top left), *bearish* (top right), *bullish* (bottom left) and *high-volatility* (bottom right).

We plot the corresponding liquidations in Figure 3. Here we observe a clear (and intuitive) dependence on the number of liquidated troves and the market condition. Indeed, notice that the number of liquidations in inversely proportional to the price of FIL. This is an intuitive result, since the CR of a given trove will decrease with the price of FIL. Interestingly enough, we see that the number of liquidated troves significantly increase as the volatility of the amrket increases.

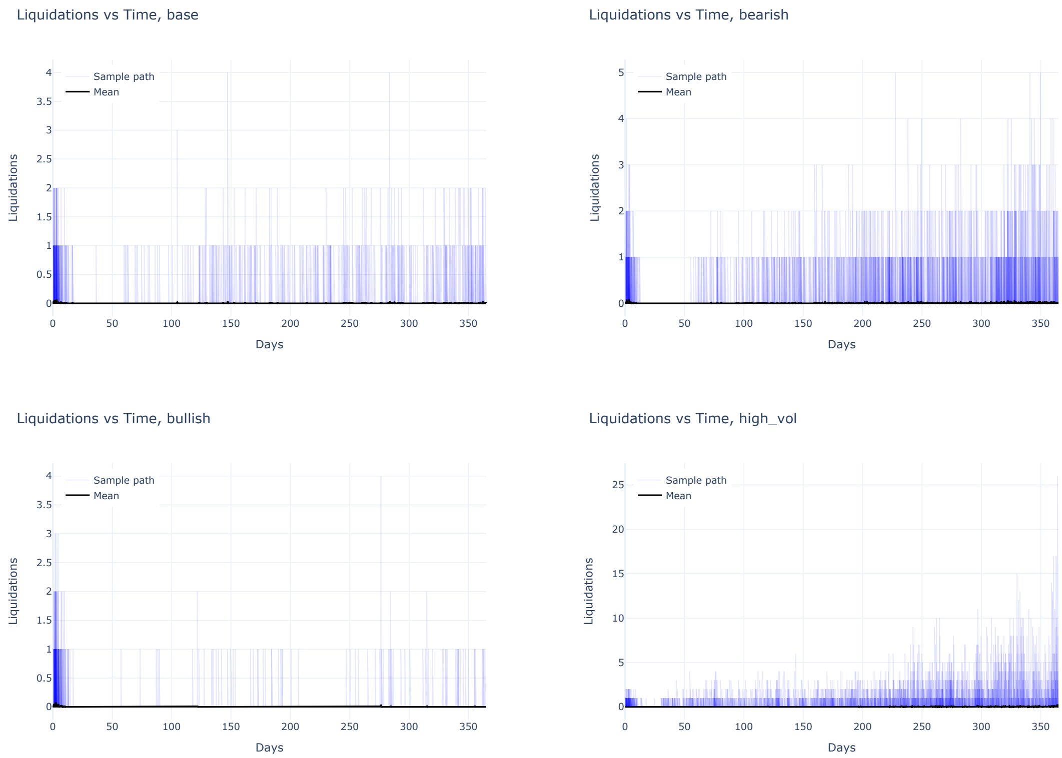

Figure 3: Liquidations vs Time under different market conditions; *base* (top left), *bearish* (top right), *bullish* (bottom left) and *high-volatility* (bottom right).

Lastly, we observe the cumulative number of open troves in Figure 4. Once again, these also have a direct dependence on the specific market condition.

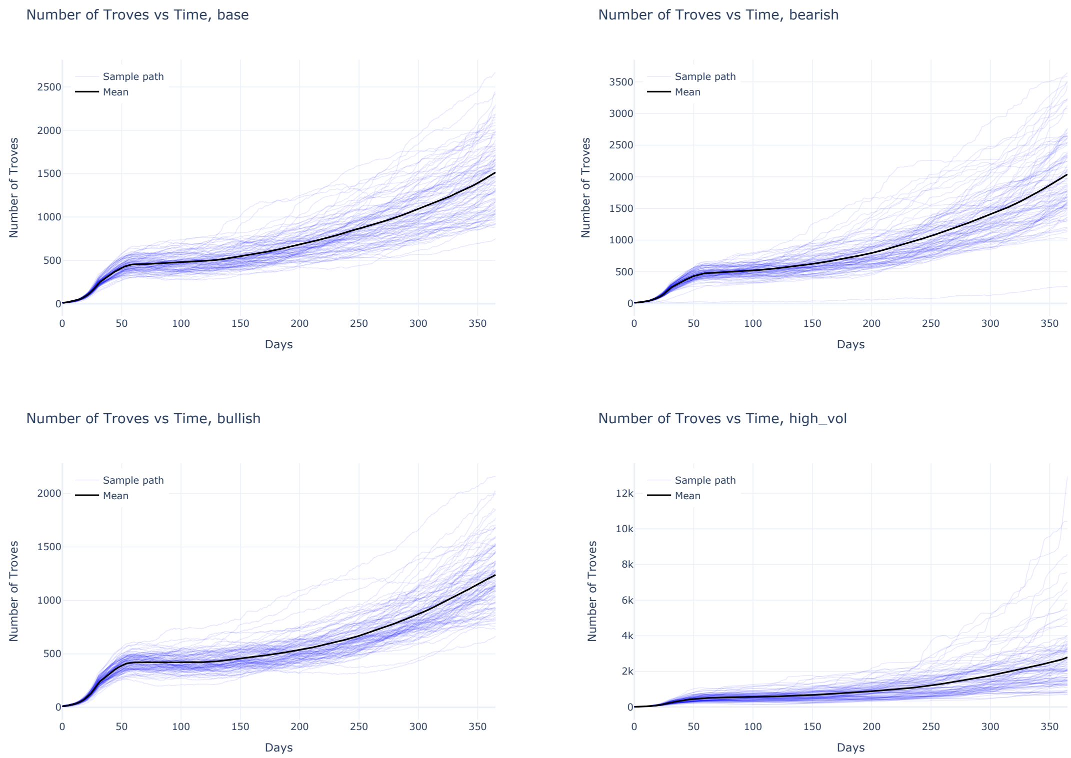

Figure 4: Cumulative number of Troves vs Time, under different market conditions; *base* (top left), *bearish* (top right), *bullish* (bottom left) and *high-volatility* (bottom right).

4 Liquidation Risk Assessment

TL;DR. *Current CR of 110% is fine to keep*

We begin by investigating potential liquidation risk. Specifically, we aim at investigating the liquidation risk, number of liquidations, etc as a function of the collateral ratio $\mathrm{CR}_{\mathrm{current}}$. Our goal is to answer the following questions:

1. *How does liquidation risk depend on CR?*

2. *Should we keep a 110% collateralization ratio, or should we use something different?*

To analyze this, we implement our simulation environment on a variety of market scenarios for FIL, with different levels of optimism, as shown below. To this end, we run our Monte Carlo simulation environment to estimate several QoI as a function of CR. Datasets can be found in [3].

Figure 5: Liquidation Probability vs CR – base case

4.1 Block Interval Analysis

We conducted a comprehensive block interval analysis to evaluate liquidation risk across different timeframes. The experiment utilized high-frequency Filecoin price data collected from 2024-04-08 to 2025-04-08, comprising 525601 distinct price points. We analyzed price movements at four different block intervals (1, 2, 10, and 100 blocks) to identify volatility patterns that could trigger liquidation events.

The methodology involved calculating the percentage price changes between blocks, then measuring the statistical distribution of these changes, and specifically calculating the probability that price drops exceed 10% (our liquidation threshold). For each interval, we compute the mean, median, standard deviation, minimum, and maximum price changes.

Table 3: Block Interval Analysis of Price Changes and Associated Liquidation Risk

Block Interval	**Mean Change (%)**	**Std Dev (%)**	**Min (%)**	**Max (%)**	**Liquidation Risk (%)**
1	-0.0055	0.3488	-22.8831	1.8047	0.019
2	0.0105	1.4375	-22.9095	98.2385	0.0381
10	0.0302	3.4635	-43.543	130.797	0.2283
100	0.2044	7.8568	-48.8413	150.253	1.1035

The results revealed that short-term price movements (1-10 block intervals) presented noticeable liquidation risk, with instances of price drops exceeding 10%. At the 1-block interval, price changes averaged -0.0055% with a standard deviation of 0.3488%, indicating relatively stable short-term price behavior but with a maximum negative movement of

-22.8831%. Similarly, 2-block and 10-block intervals showed increasing volatility with maximum negative price movements of -22.9095% and -43.543% respectively.

At the 100-block interval, we observed the highest liquidation risk in our dataset. The standard deviation increased to 7.8568%, with a minimum price change of -48.8413%, representing the largest downturn observed. This resulted in a liquidation risk of 1.1035%, indicating that approximately 1 in 91 positions would face liquidation at this interval with the current CR.

Based on the statistical properties of the observed price volatility, particularly the 100-block interval data, we can conclude that a current 110% CR should be enough for most purposes, provided that there's sufficient *liveness* in the liquidation bots (i.e., that such a liquidation is triggered within 100 blocks after dropping from 100% CR). This can confortably be managed by the protocol.

5 Multiple Scenario Analysis

Before implementing USDFC on the Filecoin network, Secured Finance designed a structured test plan in collaboration with Filecoin ecosystem stakeholders. This plan, described in the document titled *"USDFC Test Plan for Filecoin Ecosystem" (March 11, 2025)*, outlines a series of real-time scenario-driven tests intended to ensure the stablecoin's reliability and security. Each scenario addresses a specific use case or stress event, ranging from normal minting workflows and cross-ecosystem payments to extreme market crashes and oracle exploits.

The primary objectives of these tests are threefold: (1) to validate USDFC's stability mechanisms and peg maintenance in the face of volatile FIL prices; (2) to confirm smooth integration with various Filecoin-based applications (e.g., storage providers and lending markets); and (3) to analyze resilience under potential disruptions, such as rapid price drops or liquidity shortfalls.

Overall, the test plan involves a mix of interactive Zoom sessions, guided user flows, and real-time monitoring of system responses. These controlled trials highlight any weaknesses in the protocol's liquidation and redemption logic, user interfaces, or broader network integration. In the following subsections, we summarize the key findings from thirteen diverse scenarios, each representing critical aspects of USDFC's functionality under both ordinary and extreme conditions.

We implement our simulation environment to investigate the tasks described in USDFC test plan[3]. Table 4 summarizes the key metrics from 100 Monte Carlo runs for each of the 13 scenarios. The table lists the final USDFC supply, minimum observed FIL price, and a brief description of the key event triggered in each test.

[3]https://docs.google.com/document/d/1z3uGA3KmT87h1dnqnJHpz0W2npqjH1zbS8Maf2YdU74/edit?tab=t.0

Table 4: Monte Carlo Summary for 13 Scenarios

Scenario	Final Supply (USDFC)	Min FIL Price	Key Event
6	67 390	2.13	Collateral set and liquidation
7	58 443	1.55	Market crash
8	58 443	0.32	90% price crash
9	58 443	2.13	Depeg event
10	58 443	2.13	Forced redemption
11	58 443	2.13	Consecutive redemption
12	58 443	2.13	Oracle exploit
13	2 980 092	2.13	Mass minting events

5.1 Analysis

Scenario 6 shows that setting a trove at exactly 110% collateral and a subsequent slight (2%) price drop successfully triggers liquidation, indicating effective risk detection under low collateral conditions. In Scenario 7, a 50% market crash drives the FIL price down to 1.55, yet the USDFC supply remains stable, which reflects robust risk management. Scenario 8, with a 90% price crash, demonstrates extreme stress conditions, while Scenario 9 confirms that depeg events can be injected without disrupting the overall supply.

Scenario 10 tests protocol resilience under liquidity shortage, with FIL price cycling between $2.50-$5.00 while USDFC supply remains stable. No redemptions or intervention events were triggered, demonstrating adequate handling of liquidity constraints. Scenario 13 evaluates mass minting events, showing successful processing of step-function supply increases (51x greater than scenario 10), with each event adding 20,000-30,000 USDFC while maintaining consistent FIL price volatility.

Overall, the protocol exhibits reliable performance under normal and moderately stressful conditions. However, additional refinements are necessary to capture the full dynamics of lending/borrowing and to simulate complex risk factors like liquidity stress and oracle manipulation.

6 Understanding Redemption Risk

TL;DR. *Redemptions occur even at high CR. The name of the game is to have (i) a sufficiently high CR and (ii) sufficiently high liquidity in front* Redemptions in CDP protocols similar to USDFC (e.g., based on [1]) are necessary for maintaining the stablecoin peg. However, these events can create forced-liquidation risk for certain users.In order to obtain a higher understanding of these events, we begin by perfomining an analysis of the Liquidy troves in Ethereum.

Analysis of 5,758 redeemed troves shows that a total of approximately 546 million LUSD and 221,073 ETH have been liquidated. Redemptions are concentrated in troves with collateral ratios near the minimum (110%), and smaller troves are more frequently targeted. These findings highlight the need for continuous monitoring of system risk, especially under volatile market conditions.

The joint PDF analysis reveals that redemption activity primarily targets positions with lower collateral ratios (concentrated around 1.2), confirming the system's mechanism of prioritizing the most vulnerable positions first. Distinct vertical bands at specific log(LUSD Redeemed) values indicate threshold effects rather than continuous distribution. Redemption risk decreases significantly as collateral ratios exceed 2.0, with minimal activity above 3.0, validating that the economic incentives function as intended by encouraging healthier collateralization levels while systematically removing capital-inefficient positions. For users, maintaining collateral ratios above 2.0 substantially reduces redemption exposure, while protocol designers should note the discrete redemption thresholds for potential parameter optimization to enhance system stability.

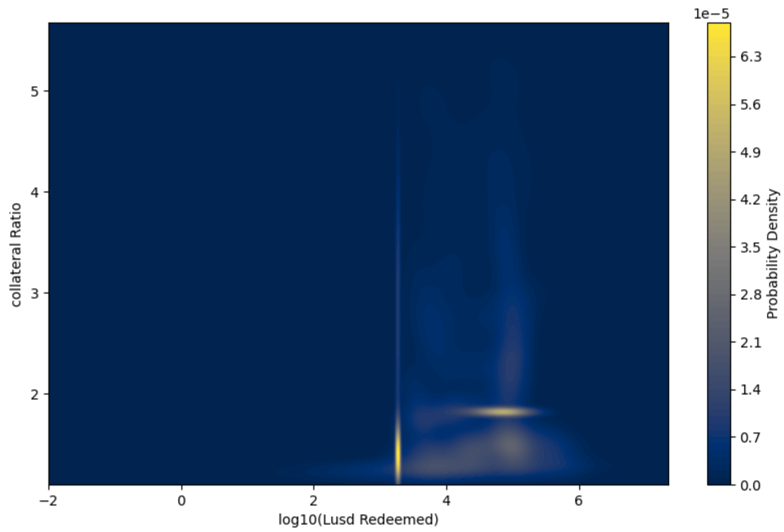

Figure 6: Joint PDF of Redemption Activity by Collateral Ratio and Position Size

6.1 Distribution of Collateral Ratios

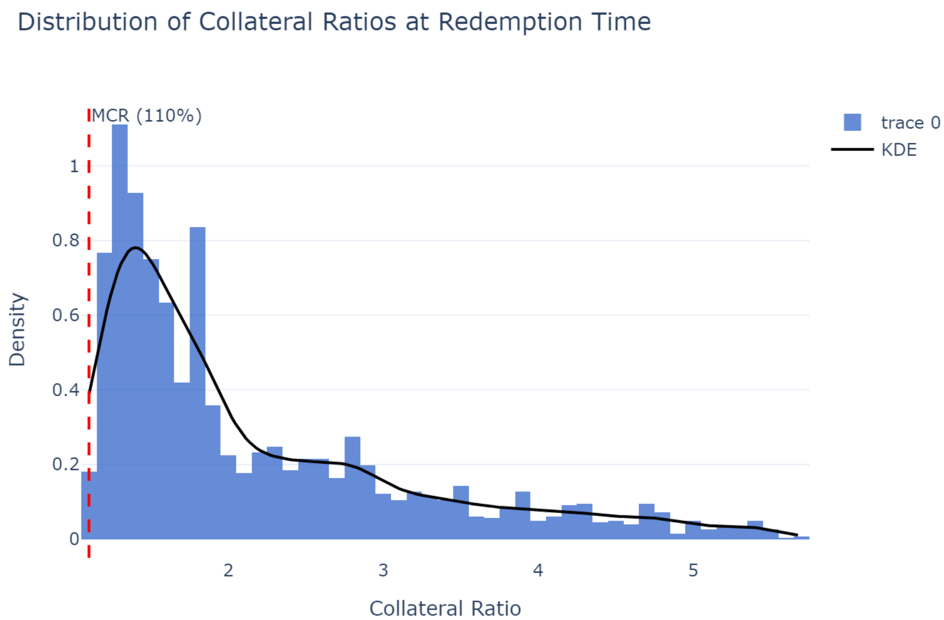

Figure 7: Collateral Ratio Distribution at Redemption

The distribution is right-skewed with a peak frequency between 1.2 and 1.5, just above the Minimum Collateral Ratio (MCR). The average collateral ratio is 2.16, and the median is 1.80, indicating that many users maintain positions close to the liquidation threshold. **Realistically, this means that, historically, redemptions can occur even at high collateral ratios.**

6.2 Collateral Size Distribution

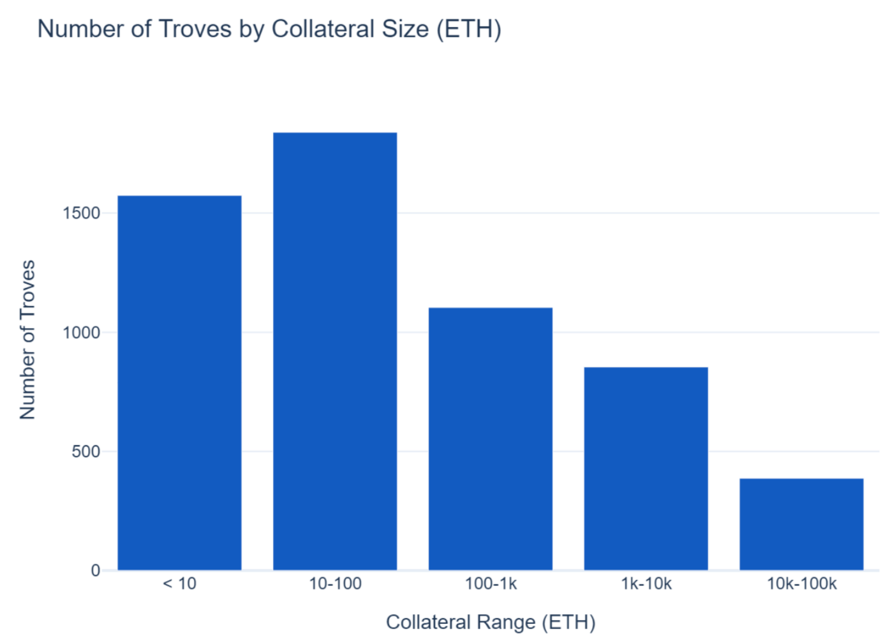

Figure 8: Collateral Size Distribution

Most troves fall in the 10–100 ETH and <10 ETH ranges, with only about 380 troves exceeding 10,000 ETH. This bimodal distribution suggests retail users maintain smaller positions while larger positions are held by institutional accounts, each carrying distinct risk profiles.

6.3 LUSD Redemption Size

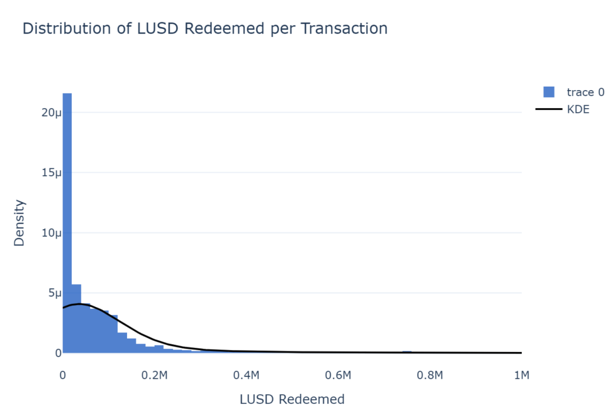

Figure 9: Distribution of LUSD Redemption Size

The redemption size distribution is highly right-skewed. The median redemption is approximately 30,364 LUSD, with the 25th percentile at 6,038 LUSD, the 75th at 94,994 LUSD, and the 99th percentile at 878,499 LUSD. **This indicates that while most redemptions are small, occasional large redemptions occur.**

6.4 Redemption Concentration

The top 10 redeemers account for roughly 39% of all LUSD redeemed. One address alone is responsible for about 62.6 million LUSD (approximately 11.5% of the total), which implies that a few large participants dominate redemption activity.

Table 5: Top 10 Redeemers by LUSD Volume

Owner Address	**Total LUSD Redeemed**
0x9c5083dd4838e120dbeac44c052179692aa5dac5	62,693,656.34
0x0561a78021d8966ddd20c28c6c4318d8675ee1f0	53,296,767.56
0x931433324e6b0b5b04e3460ef3fb3f78dda3c721	38,813,043.74
0xc2720997ea2ea9baad61e8f7de8ca3b5a1bbe1b3	13,237,524.51
0xf3f5c252e8acd60671f92c7f72cf33661221ef42	9,829,959.39
0xa1b3c586731d74178803318db709ffaac442ead7	8,885,280.76
0x5b23f5b330dfcc20b353bab85ee3a302af930005	8,506,987.53
0x1a1d3c8ded46e3f3a92dc3af8358109438d3c1c2	7,454,048.37
0xbbf2c9e6eb46a84b630f138de8f648fcecb1fad5	6,081,663.74
0x1309c007567a71b393094c21e70bd2647356a352	5,862,981.48

6.5 Collateral Ratio vs. Redemption Size

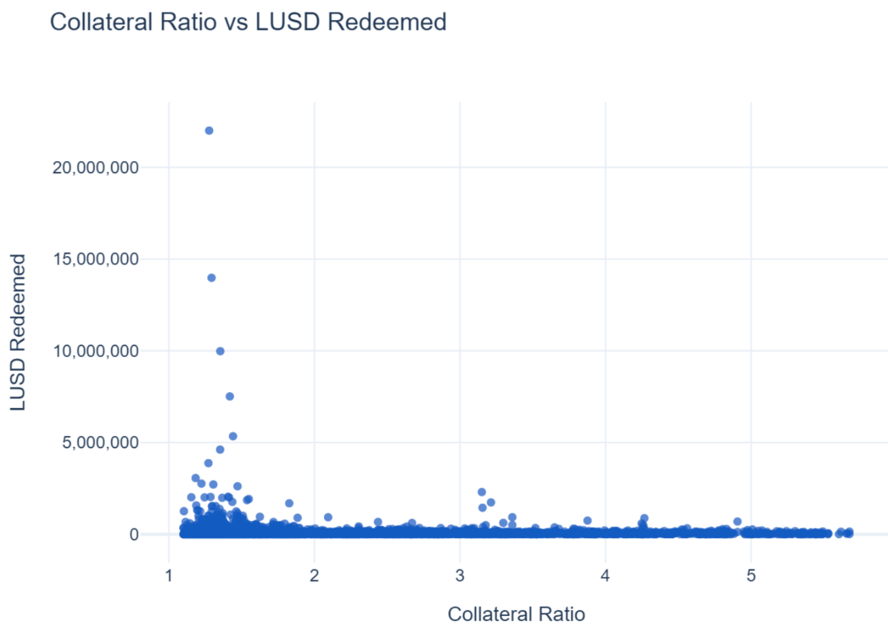

Figure 10: Collateral Ratio vs. LUSD Redemption Size

There is an inverse relationship between collateral ratio and redemption size: larger redemptions tend to target troves with collateral ratios between 1.1 and 1.5.

7 Mitigating Redemption Risk

TL;DR. *We test several mechanisms to reduce the redemption risk. We observe that all implemented methods manage to protect LP from redemption risk.*

7.1 Reducing the Likelihood of Redemptions

Reducing the likelihood of redemption can also be addressed operationally. One method involves deploying a buffer Trove with a deliberately lower collateral ratio than the LP's Trove. By setting the buffer trove's collateral ratio, CR_{buffer}, such that

$$CR_{\text{buffer}} < CR_{\text{LP}},$$

third-party redeemers naturally target the buffer trove first. This buffer trove holds a reservoir of USDFC that is used to immediately buy back FIL if redemption occurs. The acquired FIL is then reinserted as collateral, and a new borrowing is initiated—this cycle ensures that the buffer trove continuously remains the most attractive (i.e., lowest ratio) target, thereby protecting the LP's Trove from frequent redemption hits.

Implementation We use our simulation environment to test the effectiveness of this method. We run 100 Monte Carlo simulations of 1 month each. Each simulation has a scale of 1 hour. We set

$$1.1 = CR_{\text{buffer}} < CR_{\text{LP}}.$$

Our model is run on three different market condition scenarios: a *base case*, a *bearish case*, and a *bullish case*. results are shown in Tables 6-8 below.

Table 6: Base Scenario (Drift = 0.1, Vol = 0.5)

Metric	Buffer	No Buffer
$n_{\text{liquidate}}$	3.52	3.51
n_{redempt}	11.00	12.31
Final Supply	1,518,405	1,351,861
Redemption Buffer Vol	16.66	0.00
Redemption LP Vol	**0.00**	15.12

Table 7: Bearish Scenario (Drift = -0.5, Vol = 0.5)

Metric	Buffer	No Buffer
$n_{\text{liquidate}}$	6.19	6.09
n_{redempt}	7.45	9.34
Final Supply	1,505,705	1,326,649
Redemption Buffer Vol	11.76	0.00
Redemption LP Vol	**0.00**	15.04

Table 8: Bullish Scenario (Drift = 0.5, Vol = 0.5)

Metric	Buffer	No Buffer
$n_{\text{liquidate}}$	2.17	2.64
n_{redempt}	13.83	15.20
Final Supply	1,527,749	1,375,423
Redemption Buffer Vol	25.59	0.00
Redemption LP Vol	**0.00**	15.70

Across all scenarios, the introduction of the buffer trove shows clear benefits. In the base scenario, the buffer absorbs an average redemption volume of approximately 16.66 units, preventing about 15.12 units of redemption pressure from hitting LP's troves. In the bearish case, although the overall redemption activity is lower, the buffer still captures roughly 11.76 units of redemption volume compared to no absorption in the absence of a buffer—leading to fewer forced liquidations and a higher final supply (indicative of preserved collateral positions). In the bullish scenario, the buffer's impact is even more pronounced, absorbing over 25 units of redemption volume and reducing the redemption load on LP's troves.

Overall, these results suggest that a defensive buffer trove can effectively mitigate redemption risk by diverting redemptions away from LP's troves, thereby reducing the frequency of liquidations and preserving a higher final supply. This directional benefit can be critical for U.S. entities concerned with unexpected capital gains events or forced liquidations triggered by redemption waves.

7.2 Additional Liquidity

Another scenario focuses on providing sufficient liquidity to reduce depeg risk. When the stablecoin's price deviates below its $1.00 peg, it may trigger redemption waves. To counter this, liquidity pools (e.g., USDFC/FIL and USDFC/axlUSDC pools via Axelar bridging and SushiSwap on FVM) are established to maintain high liquidity and reduce price slippage. The liquidity is determined such that it can handle typical daily USDFC volumes; for instance, ensuring that a $100k trade results in less than a 1% price impact. This approach, which parallels MakerDAO's Peg Stability Module (PSM), leverages both cross-chain bridging and on-chain Automated Market Makers (AMMs) to maintain the stablecoin's peg and thereby disincentivize redemption arbitrage.

7.2.1 Actively Monitoring

Enhanced user interfaces and trove monitoring systems provide additional safeguards against redemption risk. A critical metric is the "Debt in front" of the LP's Trove, which represents the total USDFC in Troves with collateral ratios lower than that of the LP's Trove. A high "Debt in front" indicates that the LP's Trove is further down the redemption queue, reducing exposure. Conversely, if this debt diminishes, it signals an increased redemption threat. Automated alerts can notify the LP when the "Debt in front" falls below a predetermined threshold, prompting proactive measures such as adding collateral, deploying a buffer Trove, or injecting extra protocol liquidity. Tools akin to DeFiSaver can also be utilized to monitor and auto-manage trove adjustments, including partial re-collateralization, thereby reducing the risk of forced redemptions.

7.3 Tranching

One additional approach here is tranching, which is somewhat similar to the *buffering approach*. Specifically, the idea here is to modify the redemption mechanism. Consider, for the sake of the example, two *tranches* (e.g., categories) of troves: *normal* and *preferential*. Instead of executing redemptions against the troves with the smallest CR, one could do either one of these:

1. **priority**. execute redemptions against the *normal* tranches first, and if there's any liquidity left to be redeemed, execute it against the *preferential tranches*. Notice that this is quite siimlar to the *buffer* approach, but it is arguably less *manual*. This

approach protects LP from redemption risk provided that

$$\mathbb{E}\left[\mathsf{LiquidityNormalTrance}\right] > \mathbb{E}\left[\mathsf{RedeemedAmount}\right] \tag{6}$$

2. **proportional**. In this setting, the redeemable amount is split in such a way that a proportion $p \in (1/2, 1)$ is redeemed against the *normal* tranche and a proportion $1 - p < p$ gets redeemed against the preferential tranche. In this setting, LP can get *at least* a reduction of $100(1 - p)\%$ provided that (7) holds true.

7.4 Increasing Redemption Fees

One last approach is to simply increase the redemption fees. Naturally, due to the forces of supply and demand:

$$\mathbb{E}\left[\mathsf{LiquidityNormalTrance}\right] \propto \mathsf{RedemtionFees}^{-1}, \tag{7}$$

i.e., the more expensive fees are, the less incentivized people will be to redeem. This, in turn can be achieved by changing the fee mechanism, as is done in Liquity V2. That being said, after testing this mechanism we decided to not pursue it further since (i) redemptions are still an essential part of the pegging mechanism and (ii) it relies heavily on the elasticity of the redeemers.

8 Conclusions

This technical analysis shows that USDFC's core design—adapted from existing CDP-based stablecoin protocols—offers robust performance under a wide range of market conditions. By simulating the behavior of FIL-collateralized troves, monitoring liquidation thresholds, and modeling liquidity-driven price adjustments, we have demonstrated that the protocol can effectively maintain its peg against moderate and even severe price shocks.

In our stress tests, which included volatility spikes, mass redemption events, and extreme price crashes, USDFC generally exhibited resilience. Liquidations tended to occur most frequently in scenarios where collateral ratios were set too low, but adjusting the system parameters (notably the minimum collateral ratio) reduced liquidation and redemption risk considerably. Practical mitigation strategies such as buffer troves, and additional liquidity provisioning, helping to divert unwanted redemptions away from individual troves.

Although redemptions are a natural part of any collateralized stablecoin design, their impact remains manageable if the protocol is calibrated to handle rapid changes in market sentiment. By adapting parameters to Filecoin's unique pricing patterns and providing sufficient liquidity buffers, the protocol can continue operating securely without substantial risk of depegging or runaway liquidation cascades. The simulations therefore give confidence that, under well-chosen parameters and careful monitoring, USDFC's mechanism is fundamentally stable, even in stressful conditions.

References

[1] Lauko, R., Pardoe, R.: Liquity: Decentralized borrowing protocol. Tech. rep., Technical report. Available at https://docsend. com/view/bwiczmy (2021)

[2] Liu, Y.: Liquity macroeconomic model - theory results. https://colab.research.google.com/drive/1AyhFfE_EKCcMO6HeG04Se3hbraTxODWU?usp=sharing (????), google Colab Notebook. Accessed 03.20.2025

[3] Team, C.: Sf results. https://drive.google.com/drive/folders/1HKw-8zKI6TOXVr6qKG9CUOgx6SZXPwW8?usp=sharing (????), dataset. Accessed 03.20.2025

[4] Team, T.S.F.: Usdsf documentation. https://docs.secured.finance/stablecoin-protocol/introduction (2025), online Article. Accessed 03.18.2025

A Pseudocode

A.1 Subroutine: UpdateTroves

Algorithm 2 UpdateTroves(troves, t, $P_{\text{FIL},t}$, $P_{\text{USDsf},t-1}$)

for trove in troves **do**
    Compute $\text{CR}_{\text{current}} \leftarrow \frac{P_{\text{FIL},t} \times Q_{\text{FIL}}}{D}$.
    if $\text{CR}_{\text{current}} <$ threshold **then**
        LiquidateTrove(trove, $P_{\text{FIL},t}$, $P_{\text{USDsf},t-1}$)
    end if
end for
Adjust existing troves based on deviations from their initial collateral ratios.
Close troves based on random shocks and the current USDsf price.
Open new troves by sampling target collateral ratios and collateral quantities.
Compute and accumulate issuance fees from adjustments.
return updated troves and related metrics (e.g., fees, counts).

A.2 Subroutine: UpdateStabilityPool

Algorithm 3 UpdateStabilityPool($S_{\text{pool},t-1}$, t, r_{return}, D_{total})

Generate shock $\epsilon_t \sim \mathcal{N}(0, \sigma_{\text{stability}})$
Retrieve $r_{\text{natural},t}$ from the natural rate series.
Compute:

$$S_{\text{pool},t} \leftarrow S_{\text{pool},t-1} \times d \times (1 + \epsilon_t) \times (1 + r_{\text{return}} - r_{\text{natural},t})^{\theta}$$

Cap $S_{\text{pool},t}$ at D_{total}.
return $S_{\text{pool},t}$.

A.3 Subroutine: UpdateLiquidityAndPrice

Algorithm 4 UpdateLiquidityAndPrice(L_t, $P_{\text{USDsf},t-1}$, troves, t)

Update liquidity pool L_{t+1} using:

$$L_{t+1} \leftarrow L_t \times \text{drift}_L \times (1 + \text{shock}_L)$$

Compute total USDsf supply $D_{\text{total}} \leftarrow \sum_{\text{trove}} D$.

Calculate new price:

$$P_{\text{USDsf},t} \leftarrow P_{\text{USDsf},t-1} \left(\frac{L_t}{L_{t+1}} \right)^{1/\delta}$$

return L_{t+1} and $P_{\text{USDsf},t}$.

A.4 Subroutine: ArbitrageAdjustments

Algorithm 5 ArbitrageAdjustments(troves, t, $P_{\text{USDsf},t}$)

if $P_{\text{USDsf},t} >$ upper bound **then**
    Open additional troves to increase supply.
    Adjust issuance fees accordingly.
else if $P_{\text{USDsf},t} <$ lower bound **then**
    Trigger redemptions to reduce supply.
    Adjust trove parameters to reflect redemptions.
end if
return updated troves and adjusted $P_{\text{USDsf},t}$.

A.5 Subroutine: UpdateLQTYMarket

Algorithm 6 UpdateLQTYMarket(data, t, $P_{\text{LQTY},t-1}$)

if $t \leq$ month **then**
    Set $P_{\text{LQTY},t} \leftarrow P_{\text{LQTY},t-1}$ and estimate earnings via a stochastic process.
else
    Compute annualized earnings from recent issuance and redemption fees.
    Update $P_{\text{LQTY},t}$ based on:

$$P_{\text{LQTY},t} \leftarrow \text{discount factor} \times \text{PE ratio} \times \frac{\text{annualized earnings}}{\text{Total LQTY supply}}$$

end if
return $P_{\text{LQTY},t}$ and related market metrics.